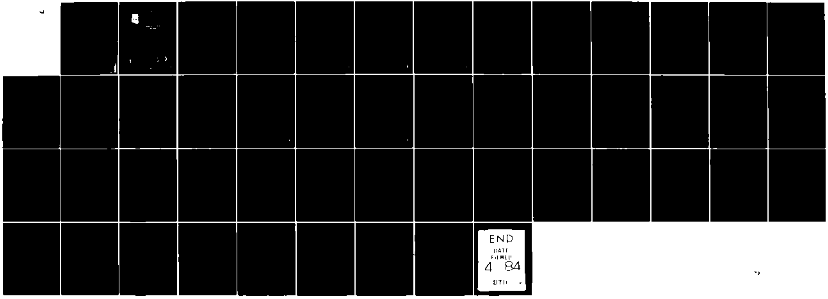
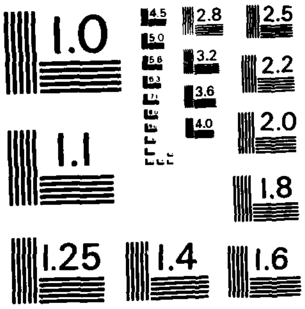
AD-A139 355 IMPLEMENTATION OF THE SIGN-LOGARITHM ARITHMETIC FFT(U) 1/1
ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND)
S J KIDD NOV 83 RSRE-MEMO-3644 DRIC-BR-90772

UNCLASSIFIED F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RSRE
MEMORANDUM No. 3644

ROYAL SIGNALS & RADAR ESTABLISHMENT

IMPLEMENTATION OF THE SIGN-LOGARITHM ARITHMETIC FFT

Author: S J Kidd

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

-a-

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3644

TITLE: IMPLEMENTATION OF THE SIGN-LOGARITHM ARITHMETIC FFT

AUTHOR: S J Kidd

DATE: November 1983

fast fourier transforms (FFT)

SUMMARY

The simulation studies described show that sign-logarithm arithmetic can be implemented in a practical digital FFT analyser. Sign-logarithm arithmetic allows a smaller wordlength than conventional fixed point arithmetic whilst maintaining performance.

Discussion of the hardware implementation of such a sign-logarithm FFT shows that power consumption can be less than conventional methods using bipolar multipliers. The use of a smaller wordlength allows a significant simplification of the system into which the FFT is placed and a higher data throughput rate.

ABSTRACT

The simulation studies described show that sign-logarithm arithmetic can be implemented in a practical digital FFT analyser. Sign logarithm arithmetic allows a smaller wordlength than conventional fixed point arithmetic whilst maintaining performance.

Discussion of the hardware implementation of such a sign-logarithm FFT shows that power consumption can be less than conventional methods using bipolar multipliers. The use of a smaller wordlength allows a significant simplification of the system into which the FFT analyser is placed and a higher data throughput rate.

RSRE MEMORANDUM 3644

IMPLEMENTATION OF THE SIGN-LOGARITHM ARITHMETIC FFT

S J Kidd

LIST OF CONTENTS

1 INTRODUCTION

Traditional hardware implementations of FFT algorithms, using fixed point arithmetic, suffer from dynamic range limitations and slow multiply speeds. The problem of multiply speed becomes more acute with the longer wordlengths required for a good dynamic range.

Sign-logarithm arithmetic attempts to overcome these problems by:

1 Increasing the dynamic range for a given number of bits by non-linear quantisation.

2 Reducing arithmetical complexity and time, since multiplication can be replaced by addition of base 2 logarithms.

The scope of this memorandum is to show how sign-logarithm arithmetic functions and how it can be implemented in a FFT analyser. The sign-logarithm FFT was simulated and compared with a fixed point FFT. A feasibility study was then conducted to determine the usefulness of this technique in a practical system.

2 GENERALISED LOGARITHM ARITHMETIC

2.1 MULTIPLICATION

The expression used for multiplication by addition of logarithms is explained in most standard algebra texts.

let $z = x * y$

$a = \log_p|x|$

$b = \log_p|y|$

$Sa = \text{sign}(x)$ where $S_k = 1,\ k > 0$

$Sb = \text{sign}(y)$ $\qquad = -1,\ k < 0$

$\qquad = 0,\ k = 0$

$m = \log_p|z| = \log_p(|x * y|)$

$Sm = \text{sign}(z) = \text{sign}(x * y)$

then

$m = \log_p|x| + \log_p|y|$

$Sm = \text{sign}(x) * \text{sign}(y)$

2.2 ADDITION AND SUBTRACTION

Addition and subtraction can be achieved when the logarithm of the two numbers is known, and an expression is easily obtained: [1]

let $s = x + y$

$a = \log_p|x|$

$b = \log_p|y|$

$Sa = \text{sign}(x)$

$Sb = \text{sign}(y)$

then $c = \log_p|s| = \log_p(|x + y|)$

$Sc = \text{sign}(s) = \text{sign}(x + y)$

(i) Assume that Sa = Sb.

then $Sa = Sb = Sc$

$$\begin{aligned} c &= \log_p(p^a + p^b) \\ &= a + \log_p(1 + p^{b-a}) \\ &= b + \log_p(1 + p^{a-b}) \end{aligned}$$

letting $d = |a - b|$

$$\begin{aligned} \text{then IF } a > b,\ c &= a + \log_p(1 + p^{-d}) \\ \text{IF } a < b,\ c &= b + \log_p(1 + p^{-d}) \\ \text{IF } a = b,\ c &= a + \log_p(2) \\ &= b + \log_p(2) \end{aligned}$$

The correction factor, $(\log_p(1 + p^{-d}))$ can only have a limited set of values; since:

$$0 < p^{-d} \leqslant 1$$

$$0 \leqslant \log_p(1 + p^{-d}) \leqslant \log_p 2$$

(ii) Assume that $Sa = -Sb$ and $a > b$

Using the same notation as before:

since $a > b$ then $|x| > |y|$

$$\begin{aligned} c &= \log_p(|x + y|) \\ &= \log_p(|x| - |y|) \\ &= \log_p(p^a - p^b) \\ &= a + \log_p(1 - p^{-d}) \end{aligned}$$

Similarly, if $b > a$ then:

$$c = b + \log_p(1 - p^{-d})$$

If, however, $b = a$, then the correct answer is $S = 0$, and a special case has to be made for the value of:

$$c = \log_p(0)$$

Addition and subtraction can be accommodated by adding a correction factor to the largest operand. The correction factor (δ) is calculated by the functions:

$$\begin{aligned} \delta = \beta(a,b) &= \log_p(1 + p^{-d}) \qquad \text{if } Sa = Sb \\ = \gamma(a,b) &= \log_p(1 - p^{-d}) \qquad \text{if } Sa = -Sb \end{aligned}$$

where $d = |a - b|$

3 FIXED POINT, BINARY SIGN-LOGARITHM ARITHMETIC

The methods developed in the previous section need modification to accommodate fixed wordlengths. In general, a binary word can be split into:

1 sign bit
p integer bits
q nominal fraction bits

for example 011.010 could represent 3.25.

In the sign-logarithm number system, a real number a is represented by its sign (Sa) and the logarithm of its magnitude L_a.

$$\text{IF } |a| \geq \frac{1}{\tau} \text{ THEN } L_a = \log_2(\tau|a|)$$

$$\text{IF } a < \frac{1}{\tau} \text{ THEN } L_a = 0$$

$$\text{IF } a < 0 \text{ THEN } S_a = 1$$

$$\text{IF } a \geq 0 \text{ THEN } S_a = 0$$

If q bits are used to represent the fractional part of the logarithm of a, L_a may be expressed thus:

$$\text{IF } |a| \geq \frac{1}{\tau} \text{ THEN } L_a = \text{ENTIER}[\frac{1}{2} + 2^q \log_2(\tau|a|)] / 2^q$$

The scale factor τ ensures positive logarithms for small numbers. The constant of 1/2 unbiases the quantisation error by rounding, rather than truncating.

Arithmetical calculations can be handled as before:

3.1 MULTIPLICATION

$$p = a * b \qquad \begin{aligned} L_a &= \log_2(\tau|a|) \\ L_b &= \log_2(\tau|b|) \\ L_p &= \log_2(\tau|p|) \end{aligned}$$

$$\begin{aligned} L_p &= \log_2(\tau|a * b|) \\ &= \log_2(\tau|a| * \tau|b| * \frac{1}{\tau}) \\ &= \log_2(\tau|a|) + \log_2(\tau|b|) - \log_2(\tau) \\ &= L_a + L_b - L_c \end{aligned}$$

where L_c is a correction factor $L_c = \text{ENTIER}[\frac{1}{2} + 2^q \log_2(\tau)] / 2^q$

The sign of the product is determined by:

$$Sp = Sa \oplus Sb$$

The expression $L_p = L_a + L_b - L_c$, allows both underflow and overflow to occur.

Underflow occurs when L_p is negative, and indicates that the correct product is smaller than the smallest sign-logarithm number, ie $p < 1/\tau$. It is useful to detect this underflow, and assign zero to the result.

Overflow occurs when the result exceeds the longest number that can be represented by an n bit binary logarithm.

$$\text{ie } L_p \geq 2^{n-q} - 1$$

Numbers can be scaled by right shifts, and doubling τ; or subtracting a constant from the logarithm.

3.2 ADDITION AND SUBTRACTION

$s = a + b$

(i) Assuming $Sa = Sb$ and $L_a > L_b$

from the generalised analysis $c = \log_p(|a + b|)$

$= a + \log_p(1 + p^{-d})$

so $L_s = L_a + \beta(L_a, L_b)$

where $\beta(L_a, L_b)$ is defined by

$$\beta(L_a, L_b) = \text{ENTIER}[\tfrac{1}{2} + 2^q \log_2(1 + 2^{-d})] / 2^q$$

$$d = |L_a - L_b|$$

The function $\beta(L_a, L_b)$ has a limited set of values in the region (0,1).

(ii) Assuming $Sa = -Sb$ and $a > b$

$L_a > L_b$

From the generalised analysis $c = \log_p(|a + b|)$

$= a + \log_p(1 - p^{-d})$

so $L_s = L_a + \gamma(L_a, L_b)$

where $\gamma(L_a, L_b)$ is defined by

$$\gamma(L_a, L_b) = \text{ENTIER}[\tfrac{1}{2} + 2^q \log_2(1 - 2^{-d})] / 2^q$$

$$d = |L_a - L_b|$$

Since $d \geqslant 0$, a special case exists when $d = 0$, $L_a = L_b$.

IF $d = 0$ THEN $L_s = 0$

The sign of the solution is given by the largest value, in this case $Ss = Sa$.

4 LOGARITHMIC QUANTISATION

Non-linear quantisation is frequently used in speech communication systems to improve dynamic range and to maintain a constant signal to quantisation noise power ratio.

Any system of sign-logarithm arithmetic requires quantisation of the signal in logarithmic intervals. Thus the advantages of non-linear quantisation also apply to any system using sign-logarithm arithmetic.

A first order approximation for the quantisation noise for a sinusoid is as follows.

4.1 LINEAR QUANTISATION

Betts [5] gives the quantisation noise power for a signal constrained in the interval ±v and quantised to m levels of equal spacing (Δv) as:

$$Nq = \frac{\Delta v^2}{12} = \frac{v^2}{3m^2} \qquad \text{since } \Delta v = \frac{2v}{m}$$

Signal power $S = v^2\sigma^2$ where σ^2 is the signal variance

$$\therefore \frac{S}{Nq} = 3m^2\sigma^2$$

which is dependent upon the number of levels and the signal variance

4.2 NON-LINEAR QUANTISATION

A typical logarithm quantising law is of the form:

$$\begin{aligned} y &= K\,\mathrm{Ln}(\tau|x|) && x \geqslant \tau \\ &= 0 && |x| < \tau \\ &= -K\,\mathrm{Ln}(\tau|x|) && x < -\tau \end{aligned}$$

where K and τ are scaling factors.

Such a quantising law is shown in Figure 1, for the case of an 8 bit logarithm with three fractional bits.

$$\tau = 6 \times 10^4$$

If $y = K\,\mathrm{Ln}(\tau x)$ for $x \gg \tau$

then $\frac{dy}{dx} = \frac{K}{x}$

$$\approx \frac{\Delta y}{\Delta x}$$ where Δx and Δy represent intervals in x and y, Δy the quantisation interval.

$$\Delta x = \frac{x}{K}\Delta y$$

$$= \frac{2x}{Km}$$ since $\Delta y = \frac{2}{m}$ if m is the total number of levels, and the signal is constrained in the interval ±1.

Betts [5] gives the quantisation noise power as:

$$Nq = \frac{1}{12}\sum_j (\Delta x_j)^2 p(x_j)\Delta x$$ for j levels, with the probability that the signal is in the jth interval $p(x_j)$

$$\approx \frac{1}{12}\int_{-1}^{1} (\Delta x)^2 p(x)$$ for large m

p(x) is the probability density function of the signal x.

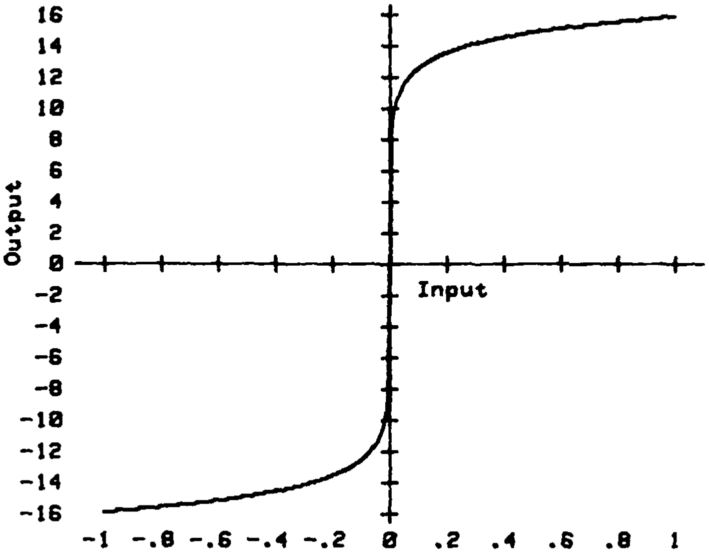


Figure 1. LOG Quantisation law
(8 bits, 3 bit fraction)

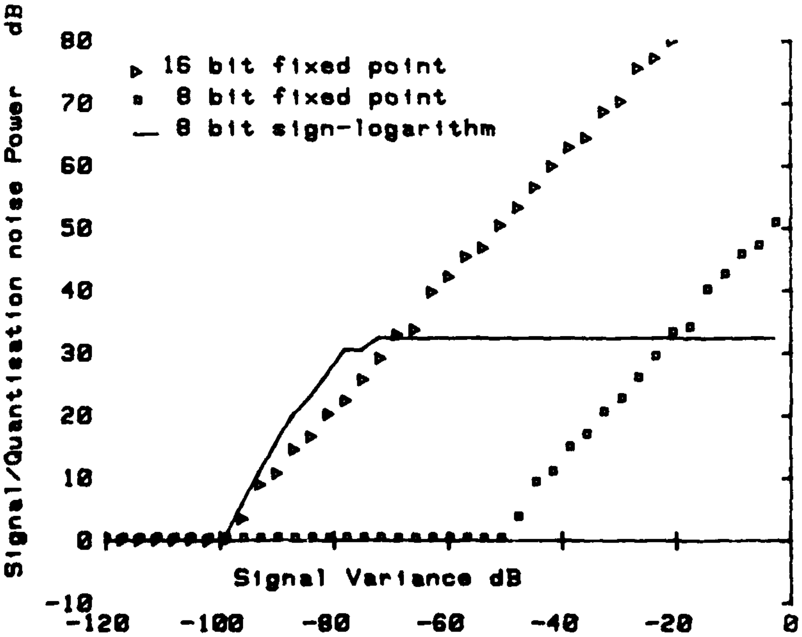


Figure 2. Signal/Quantisation noise ratio
vs. Signal variance.

Substituting for Δx where $x \gg \tau$ and large m gives

$$Nq = \frac{1}{12}\int_{-1}^{1}\left(\frac{2x}{Km}\right)^{2} p(x)dx$$

$$= \frac{1}{3K^{2}m^{2}}\int_{-1}^{1} x^{2}p(x)dx$$

$$= \frac{\sigma^{2}}{3K^{2}m^{2}}$$ since we have assumed that th gnal is wholly contained in the inter

Thus $\frac{S}{Nq} = \frac{\sigma^{2}3K^{2}m^{2}}{\sigma^{2}} = 3K^{2}m^{2}$ for $x > \tau$

and is independent of the signal.

A more detailed analysis, which accounts for all signal levels, is shown in Figure 2 (with $\tau = 6 \times 10^{4}$ for the logarithm quantisation). It can be seen that the signal to quantisation noise power for logarithmic quantisation remains constant over a much wider range than linear quantisation.

It may therefore be assumed that for a system which encounters a wide variety of signal levels, logarithm quantisation should be used.

5 SIMULATION

5.1 METHOD OF ANALYSIS

A sign-logarithm Radix-2 FFT processor was simulated on a computer to verify this technique. The simulation was based upon the rules described in previous sections. The sign-logarithm processing was compared with a simulated fixed point Radix-2 FFT, for different wordlengths. Both FFTs automatically rescaled their operands when it was likely that an operation would result in an overflow. Hamming windowing was included in the processing of both FFTs.

The output from the sign-logarithm FFT was expanded so that a reasonable comparison could be made (compression removed). The ADC clipping levels were set at ±1 V. The peak signal amplitude was 0.25 V, with a signal to noise ratio of 40 dB.

Results were displayed graphically.

5.2 SIMULATION RESULTS

The results are described briefly in this section, but are shown in more detail in Appendix A.

(i) For a fixed wordlength, the number of bits used for the fractional part of the logarithm affects the sidelobe levels of the sign-logarithm FFT. An optimum number of fractional bits exists for each wordlength.

WORDLENGTH	OPTIMUM NUMBER OF FRACTIONAL BITS
8	3
6	2
5	1

(ii) For the wordlengths used, and with an optimum number of fractional bits. the sign-logarithm FFT gave superior results to the fixed point FFT with the same wordlength. This may be attributed to the increased dynamic range.

5.3 DISCUSSION OF RESULTS

From the simulation, two implications exist:

(i) For the same wordlengths, superior results may be obtained using a sign-logarithm FFT.

(ii) The same performance as a fixed point FFT may be achieved with a smaller wordlength, by using sign-logarithm arithmetic.

The ability to add logarithms in place of multiplication, coupled with the improved performance of the sign-logarithm FFT, indicates that this method could offer significant advantages in processing time and power dissipation over conventional methods

6 HARDWARE IMPLEMENTATION

6.1 ARITHMETIC UNITS

The radix-2 butterfly structure requires four real multipliers and six add/subtract units (Figure 3). A radix-2 butterfly for sign-logarithm arithmetic may be constructed from hardware which follows the arithmetical rules established in section 3, assuming that the data has been quantised in a logarithmic fashion as described in section 4.

A multiplier requires the addition of the logarithms according to the rules

$$L_p = L_a + L_b - L_c$$

$$S_p = S_a \oplus S_b \qquad (3.1)$$

The required hardware is shown in Figure 4, which also deals with underflow for small numbers. Since one operand is a weighting function stored in ROM, the weighting function may be adjusted prior to addition so that it includes the correction factor L_c.

An add/subtract unit requires a more complex arrangement shown in Figure 5. The correction factors $\gamma(L_a,L_b)$ and $\beta(L_a,L_b)$ are modified to:

$\gamma(d)$, $\beta(d)$

enabling the coefficients to be stored in a small ROM and addressed by the difference between L_a and L_b.

6.2 RADIX-2 BUTTERFLY

A radix-2 butterfly may be constructed from the units described in 6.1 as shown in Figure 6. Some redundancy has enabled the component count to be reduced. Scaling of the results may be achieved by adjusting the look up tables in the final add/subtract units as shown.

6.3 REQUIREMENTS AND CONSEQUENCES OF IMPLEMENTATION

6.3.1 COMPARISON OF SPEED AND POWER

Comparisons were made on the basis that sixteen bit fixed point FFTs are equivalent to eight bit sign-logarithm FFTs. The figures shown are for the parallel implementation of one butterfly shown in Figure 3.

a Speed

The longest propagation delay in the radix-2 butterfly is the time required to complete one multiply and two adds. The sign-logarithm implementation required five adds and two ROM delays (Figure 6).

bipolar	16 bit		100 + 26 + 26	=	152 ns
cmos	16 bit	(TMC 216H)	170 + 26 + 26	=	222 ns
cmos	16 bit	(ADSP 1016)	130 + 26 + 26	=	182 ns
bipolar	16 bit	(MPY 016K)	45 + 26 + 26	=	97 ns
bipolar	8 bit	sign-logarithm	5 x 20 + 2 x 30	=	160 ns

b Power

Calculations are based on the typical power consumptions for all the components required to implement one butterfly. This requires:

conventional implementation	- 4 multipliers, 6 x 16 bit adders.
sign-logarithm implementation	- 14 x 8 bit adders, 4 x 128 x 8 ROMs. 4 x 8 bit MUX, 4 x 8 bit comparators.

bipolar	16 bit		30 W
cmos	16 bit	(TMC 216H)	14 W
cmos	16 bit	(ADSP 1016)	12.6 W
bipolar	16 bit	(MPY 016K)	30 W
bipolar	8 bit	sign-logarithm	16 W

X = A + B*EXP(j2*PI*n/N)
Y = A - B*EXP(j2*PI*n/N)

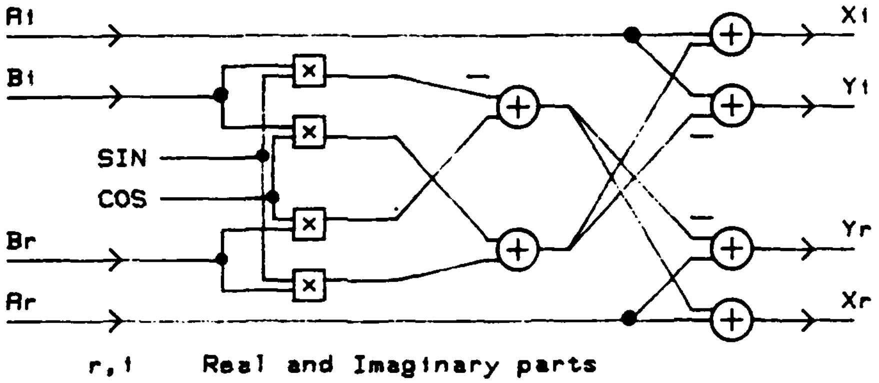


Figure 3. Radix-2 Butterfly, real and imaginary parts.

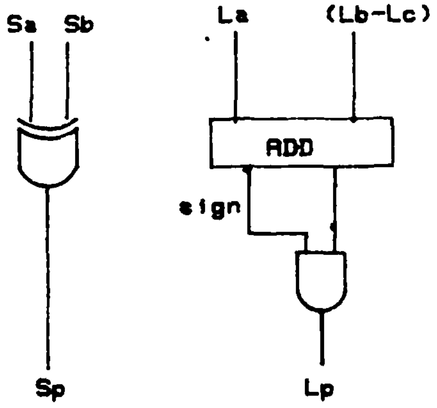


Figure 4. Hardware implementation of Sign-logarithm multiplier.

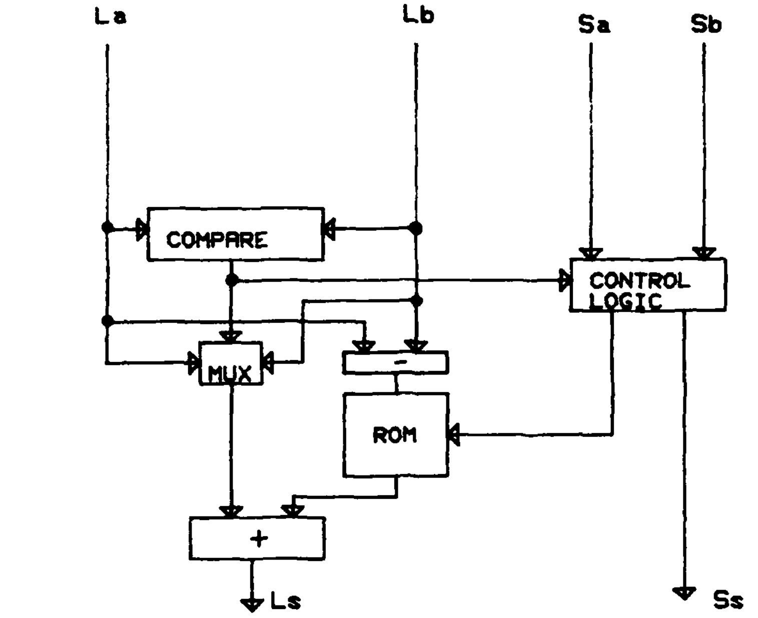


Figure 5. Hardware implementation of sign-logarithm Adder

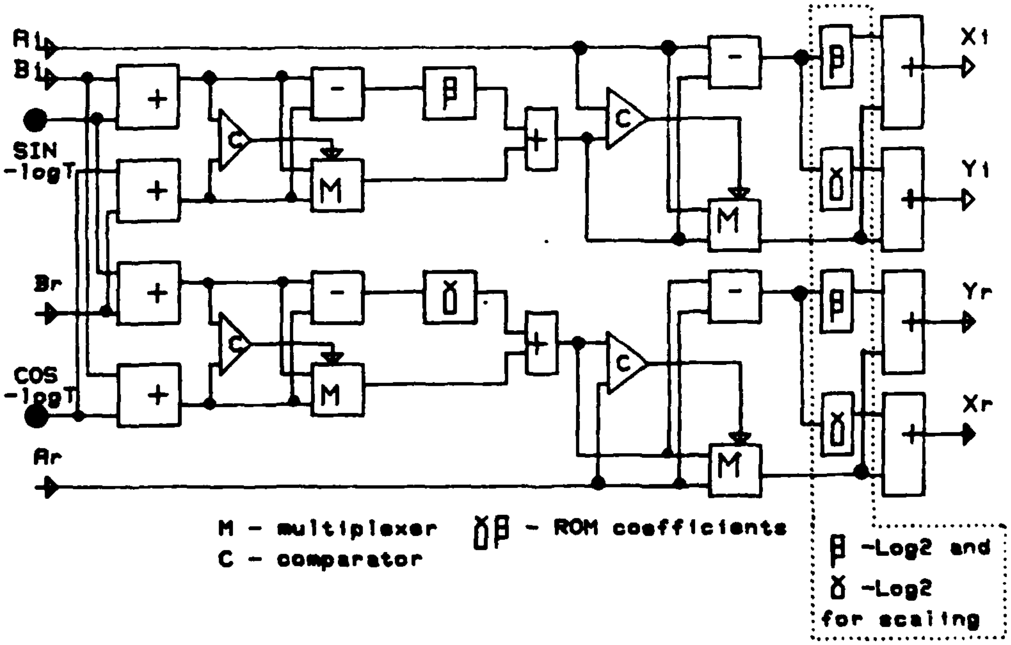


Figure 6. Sign-logarithm Radix-2 Butterfly

6.3.2 DISCUSSION

The gain in processing time made by adding logarithms for multiplication is offset by the more complicated procedure required to add two numbers together. Whilst the sign-logarithm butterfly consumes less power than its equivalent bipolar fixed point butterfly, this advantage has been 'leapfrogged' by fast cmos multipliers.

The sign-logarithm FFT is likely to be cheaper to implement whilst the cost of multiply chips remains high.

The logarithmic quantisation may be achieved by preceding an ADC with a non-linear amplifier with the required transfer characteristic.

Consideration of a complete system into which a sign-logarithm FFT is placed shows several advantages.

a ADCs with fewer bits can be used. These are faster and consume less power.

b Smaller wordlengths generally allow faster data throughput rates in all circuits.

c Smaller wordlength allows simpler board layouts, but additions increase the complexity.

d The greater dynamic range available with non-linear quantisation allows earlier analogue to digital conversion in a system. The need for filtering to reduce dynamic range before conversion is eliminated.

7 CONCLUSIONS

The sign-logarithm FFT is an attractive alternative to conventional fixed point implementations. It offers:

a greater dynamic range for an identical number of bits

b almost constant signal to quantisation noise power ratio

c a reduction in power consumption when compared to bipolar multiplier configurations

d [illegible]ller wordlengths for similar performance

e [illegible]fication of preceding and subsequent signal processing because [illegible]maller wordlengths

Th[illegible]eved with a slight increase in the time required to perform a radix-2 butte[illegible], a consequence of the more complicated procedure required to add two numbers together.

8 APPENDIX A

SIMULATION RESULTS

The results of computer simulation of fixed point FFTs and sign-logarithm FFTs with different wordlengths are shown in Figures A1 to A28.

The figures are shown in two formats.

a An annotated contour plot of the modulus output of 15 x 16 point complex FFTs. Points exist on the plus marks, contours are interpolated between them at -3 dB intervals. Plots are shown for many different word configurations.

b A graph of the modulus output of one or more 16 point FFTs for differing word configurations.

KEY TO FIGURES

A1 Contour plot with 16 bit fixed point arithmetic.

A2 Contour plot with 8 bit fixed point arithmetic.
Not significantly different to A1 down to -30 dB.

A3 } Contour plot with 8 bit sign-logarithm arithmetic.
A4 } Optimum number of bits for the fractional part is 3
A5 } and shows no significant difference to A2 down to -30 dB.

A6 Contour plot with 6 bit fixed point arithmetic.
Sidelobe levels higher than -24 dB due to the
inability to resolve the full dynamic range required.

A7 } Contour plot with 6 bit sign-logarithm arithmetic.
A8 } Optimum number of fractional bits is 2.
A9 } A significant improvement in sidelobe levels over A6.

A10 5 bit fixed point arithmetic.
Sidelobe levels higher than -18 dB.

A11 } 5 bit sign-logarithm arithmetic.
A12 } Showing optimum number of fractional bits is 1.
A13 } A significant improvement in sidelobe levels over A10.

A14 } 5 bit arithmetic, showing an 18 dB improvement in
A15 } dynamic range available with sign-logarithm arithmetic,
with no loss of precision.

A16 } Showing a significant improvement in dynamic range using
A17 } 8 bit sign-logarithm arithmetic over 8 bit fixed point.
A18 } Performance of the 8 bit sign-logarithm FFT is comparable
to the 16 bit fixed point FFT.

16 bit fixed point vs 8 bit sign-logarithm

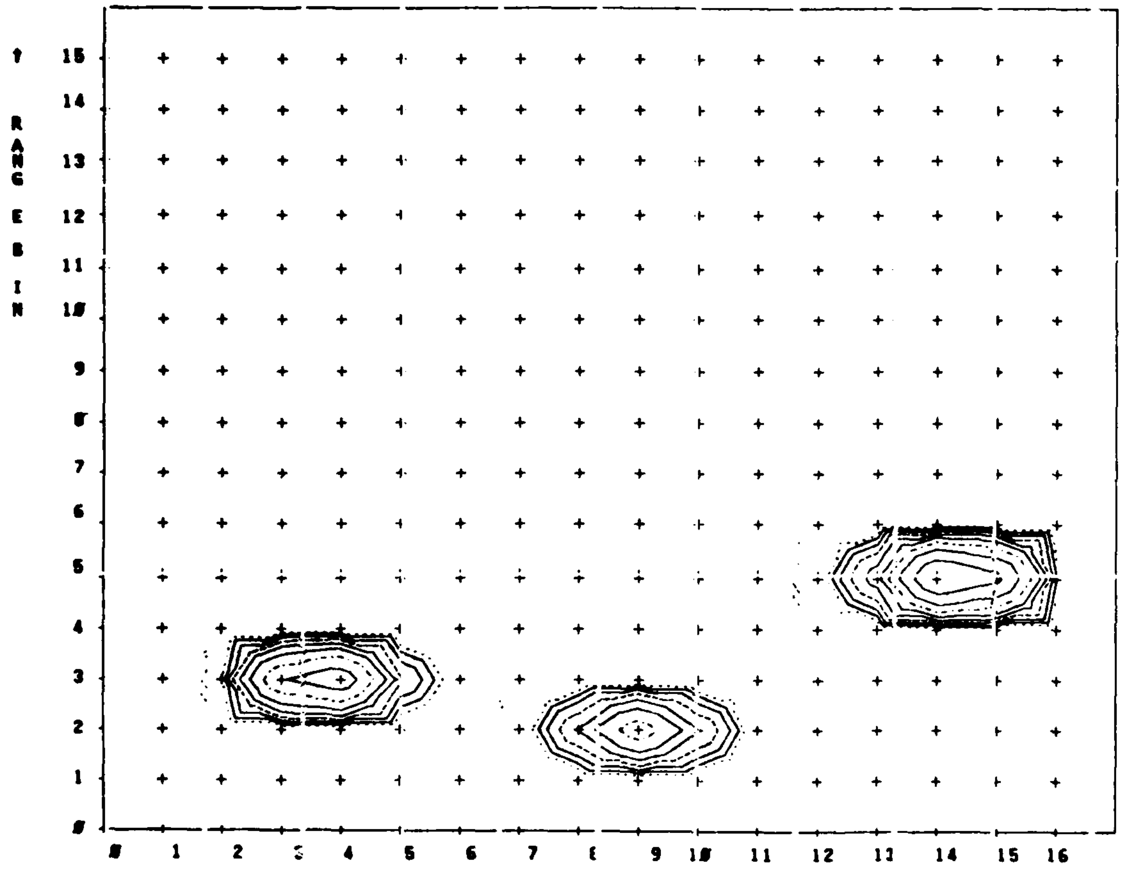
A19 Contour plot and graphs of the modulus output of
A20 16 bit fixed point FFT with more than one frequency
A21 component.
A22
A23

A24 Contour plot and graphs of the modulus output of
A25 8 bit sign-logarithm FFT with more than one
A26 frequency component.
A27
A28

The two systems do not differ significantly in performance.

```
PRODUCED BY ':WI.S:K-VT32' ON 28OC183 AT 11.5  ;5      1


                RADAR HARDWARE SIMULATION   + SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

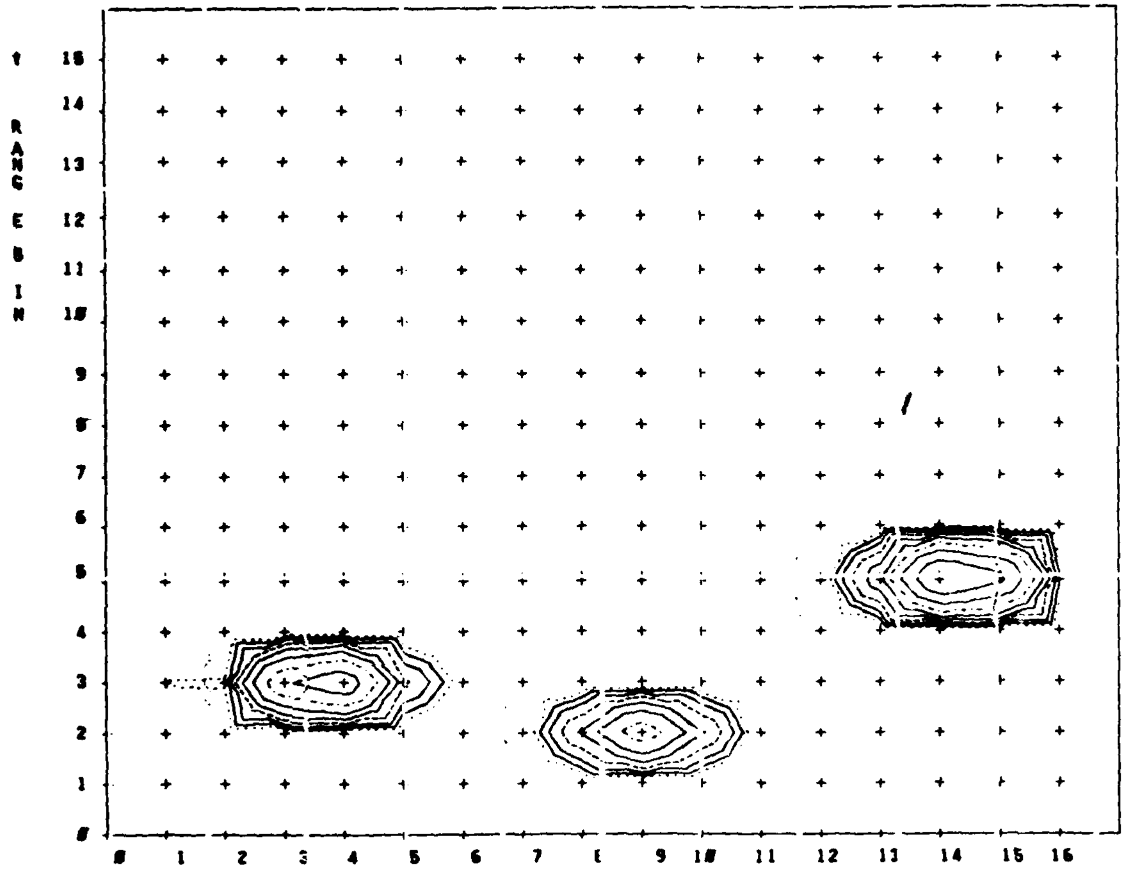


```
WINDOW DATA
( WINDOW        CALCULATION
(MANT   EXP     MANT     EXP
 16      Ø       16       Ø

   SIDELOBES(DB:      TYFE
      4Ø.Ø              2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR


FFTDATA
           CALCULATION
 MANT(INT)        EXP(INT:
   16               -1

         FFT WEIGHTS
   MANT(INT)      EXP(INT:
     16             -1
```

Figure A1

RADAR HARDWARE SIMULATION + SIGNAL PROCESSING OUTPUT

CONTOUR HEIGHTS ARE 3DB INTERVALS

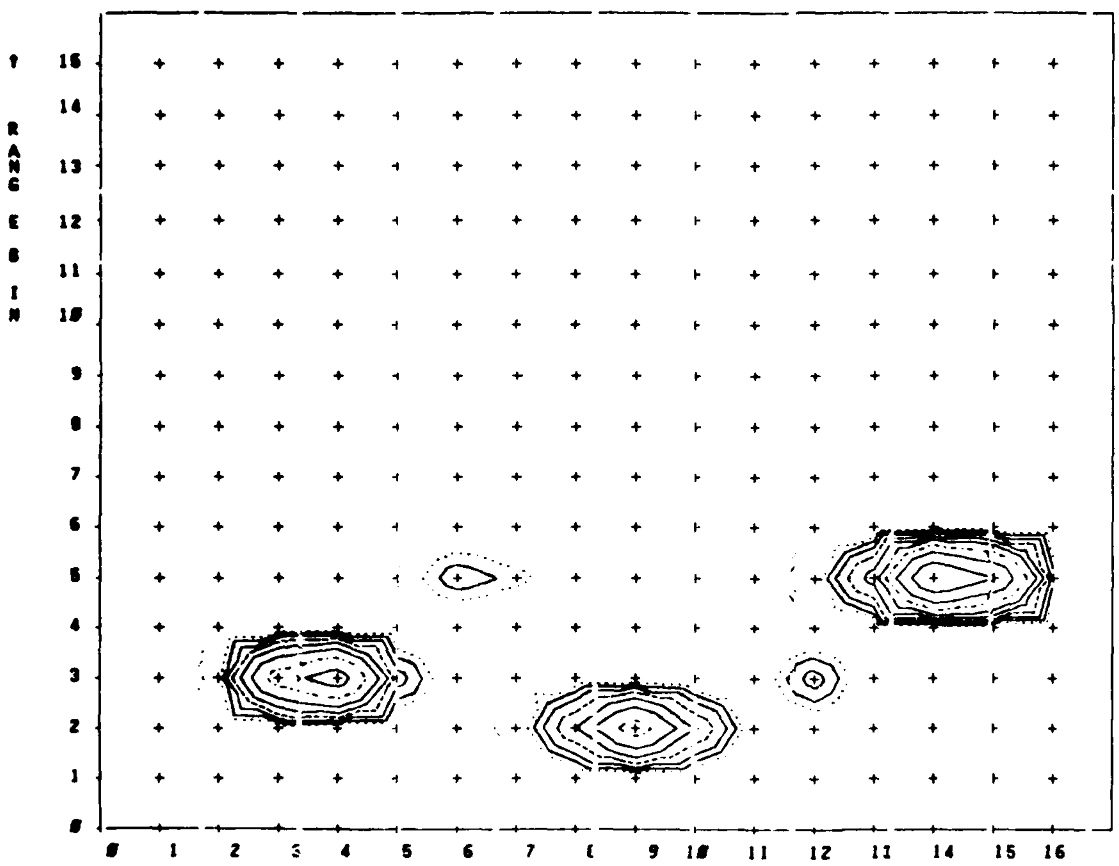


```
WINDOW DATA
  WINDOW            CALCULATION
 MANT   EXP         MANT      EXP
  8      Ø           8         Ø

     SIDELOBES(DB)        TYPE
        4Ø.Ø               2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR


FFTDATA
            CALCULATION
   MANT(INT)        EXP(INT)
      8                -1

            FFT WEIGHTS
    MANT(INT)       EXP(INT)
       8               -1
```

Figure A2

```
PRODUCED BY '1W1.S K-VT2A' ON 26OCT83 AT 1 .2  .7      1

   RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS



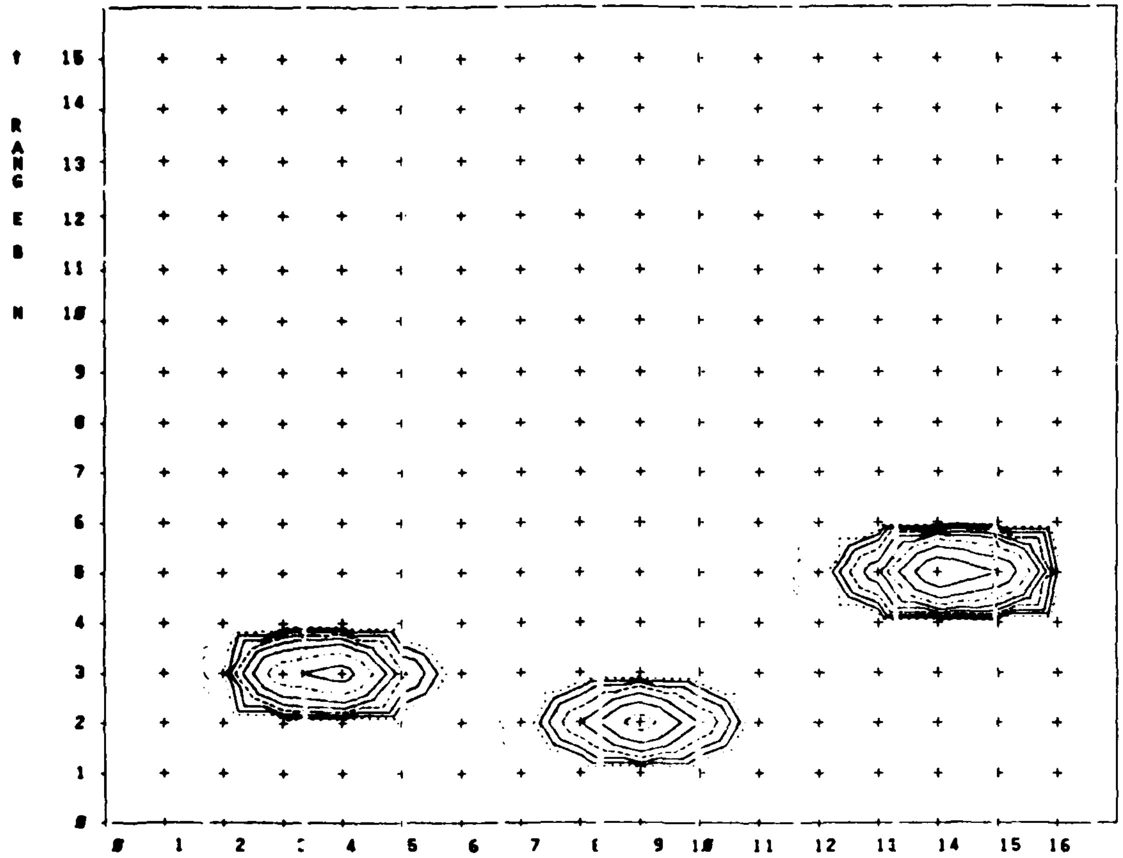
```
LOGARITHMC  PROCESS  DATA
  WINDOW        CALCULATION
 BITS   FRAC    MANT     EXP
  8      2       8        0

   SIDELOBES(DB)      TYPE
      40.0              2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A3

```
PRCDUCED BY '↑WI.SCK-VT2A' ON 26OC183 AT 15.2   6      1
```

RADAR HARDWARE SIMULATION - SIGNAL PROCESSING OUTPUT

CONTOUR HEIGHTS ARE 3DB INTERVALS

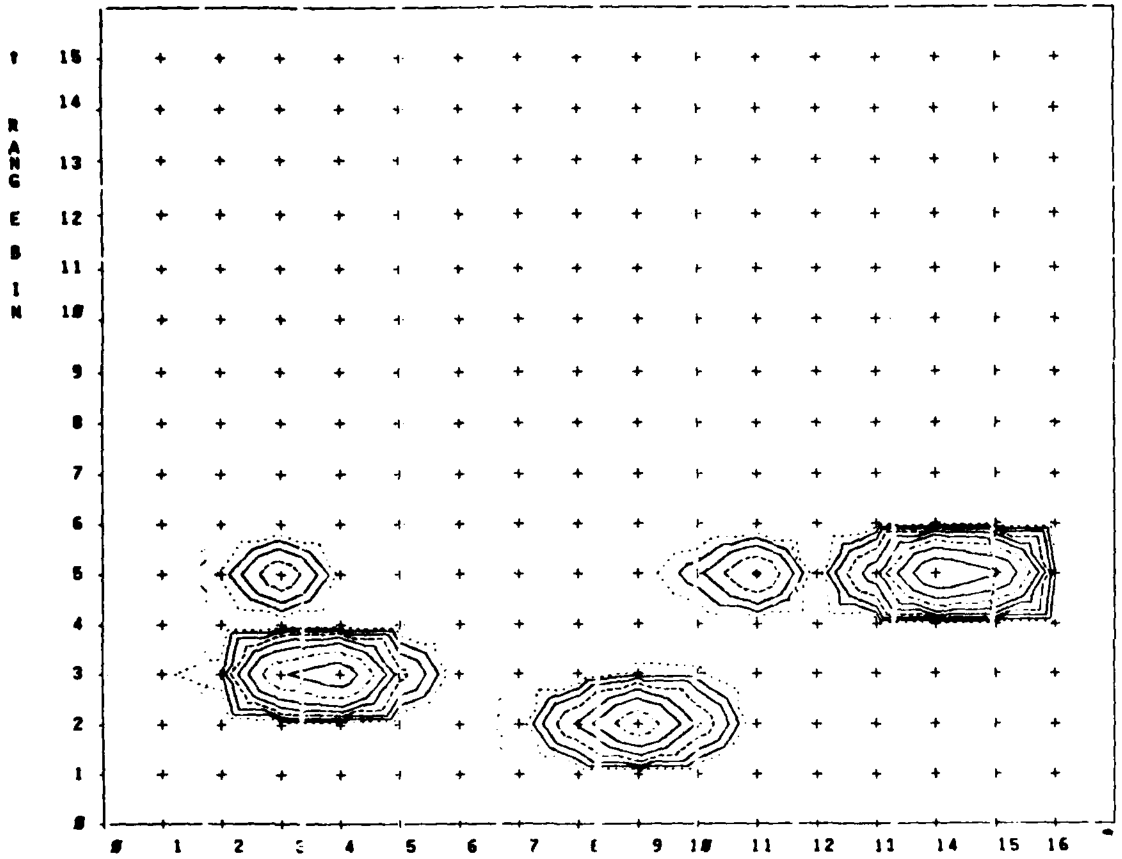


```
LOGARITHM  PROCESS  DATA
 WINDOW          CALCULATION
BITS    FRAC     MANT      EXP
 8       3        8         0

   SIDELOBES(DB)       TYPE
      40.0              2

1-UNIFORM

2-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A4

```
PRODUCED BY '1W1.SCK-VT2A' ON 26OC183 AT 1E.2  /2      1


                         RADAR HARDWARE SIMULATION  +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

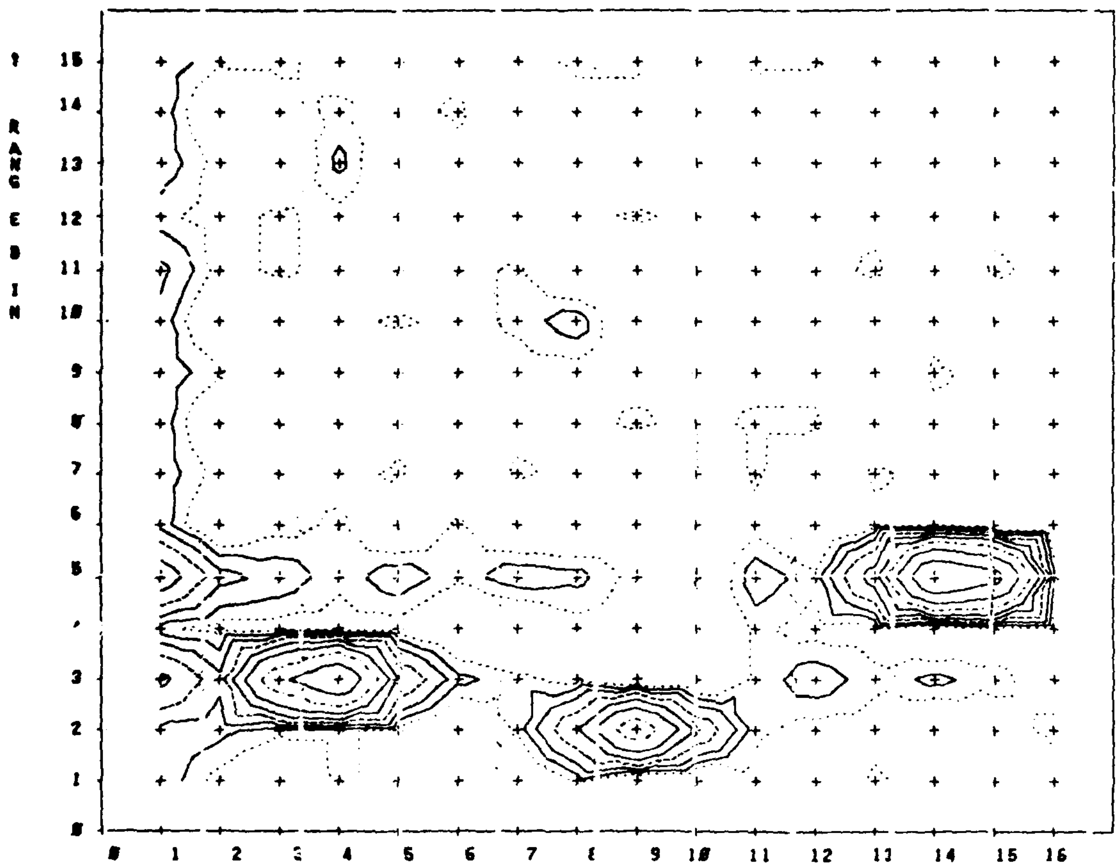


```
LOGARITHM  PROCESS  DATA
  WINDOW        CALCULATION
 BITS    FRAC     MANT     EXP
  8        4        8       Ø

    SIDELOBES(DB)       TYPE
        4Ø.Ø               2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A5

```
PRODUCED BY 'W1.SCK-VT2A' ON 27OCT83 AT 15.2 .9        1


                    RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

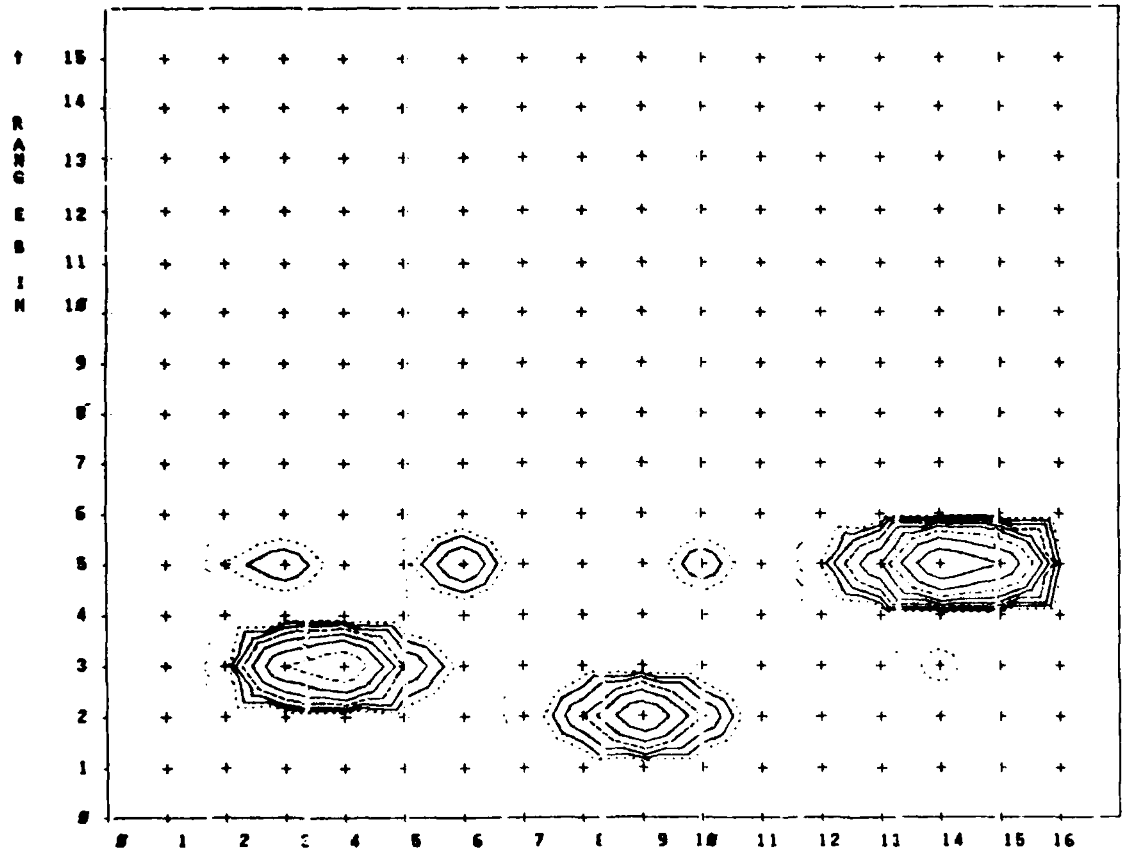


```
WINDOW DATA
  WINDOW           CALCULATION
 MANT   EXP        MANT       EXP
  6      Ø          6          Ø

     SIDELOBES(DB)        TYPE
        4Ø.Ø                2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR


FFTDATA
          CALCULATION
  MANT(INT)        EXP(INT)
      6               -1

          FFT WEIGHTS
   MANT(INT)       EXP(INT)
      6               -1
```

Figure A6

```
PRODUCED BY ':W1.S.K-VT2A' ON 26OC183 AT 1E.1  2      1

             RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

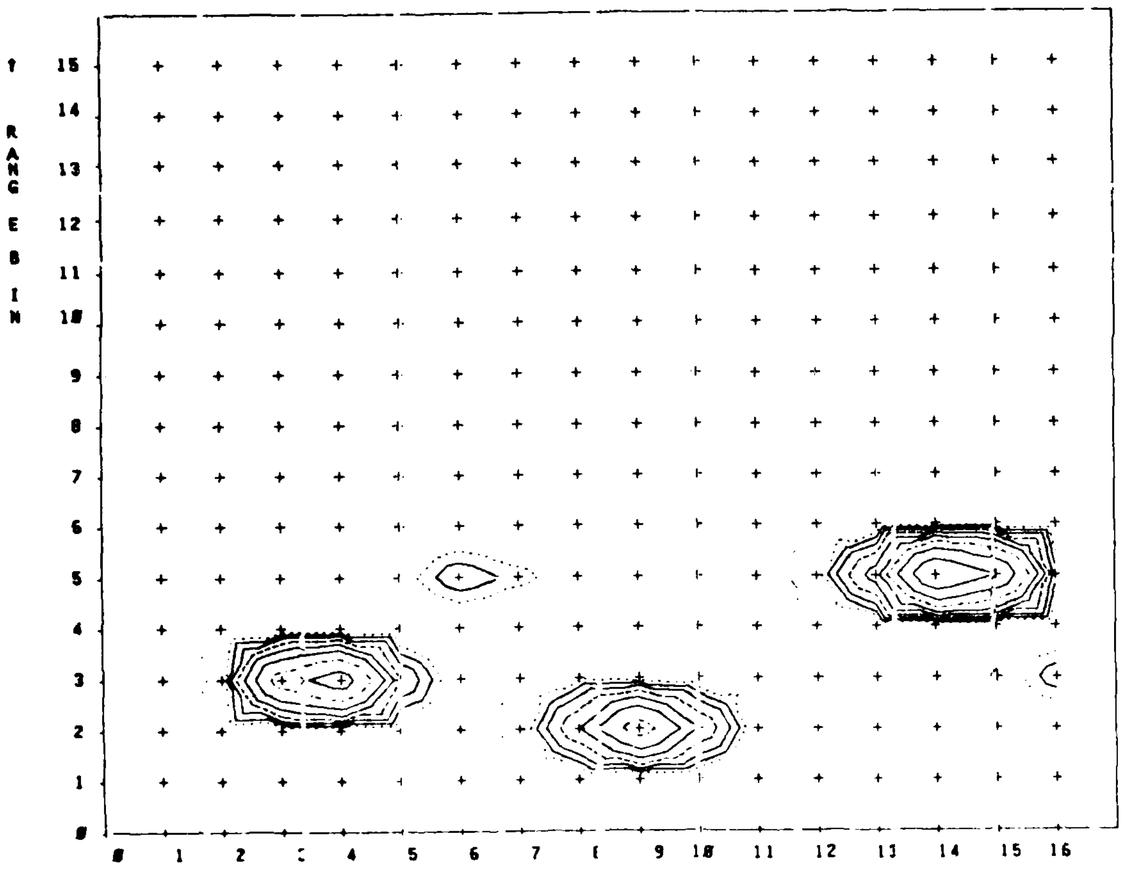


```
LOGARITHM  PROCESS  DATA
  WINDOW        CALCULATION
 BITS   FRAC    MANT      EXP
  6      1       6         Ø

   SIDELOBES(DB:      TYPE
      4Ø.Ø              2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A7

```
PRODUCED BY 'TWI.SCK-VT2A' ON 26OCT83 AT 15.1  .0       1


                     RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

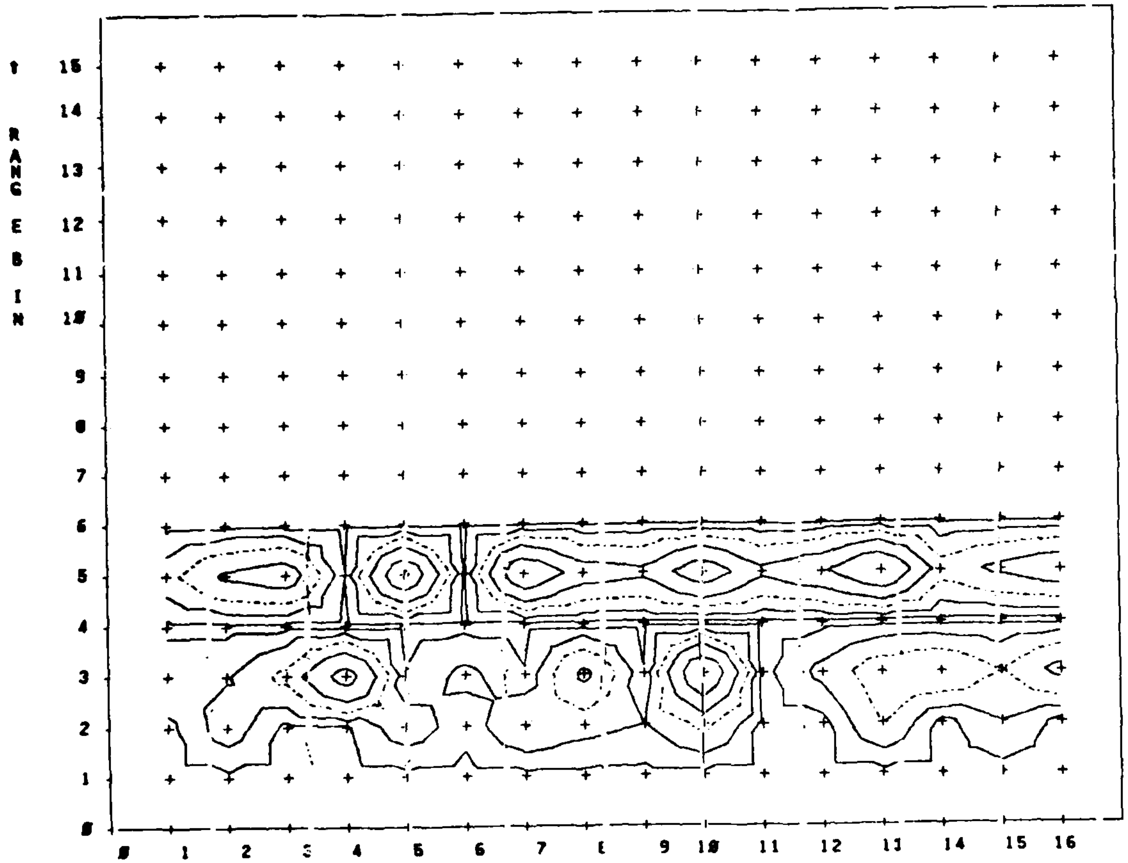


```
LOGARITHM  PROCESS  DATA
   WINDOW          CALCULATION
 BITS    FRAC     MANT       EXP
  6        2        6         0

   SIDELOBES(DB)        TYPE
        40.0               2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A8

```
PRODUCED BY ':W1.S:K-VT2A' ON 26OC183 AT 15.1  6      1


              RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

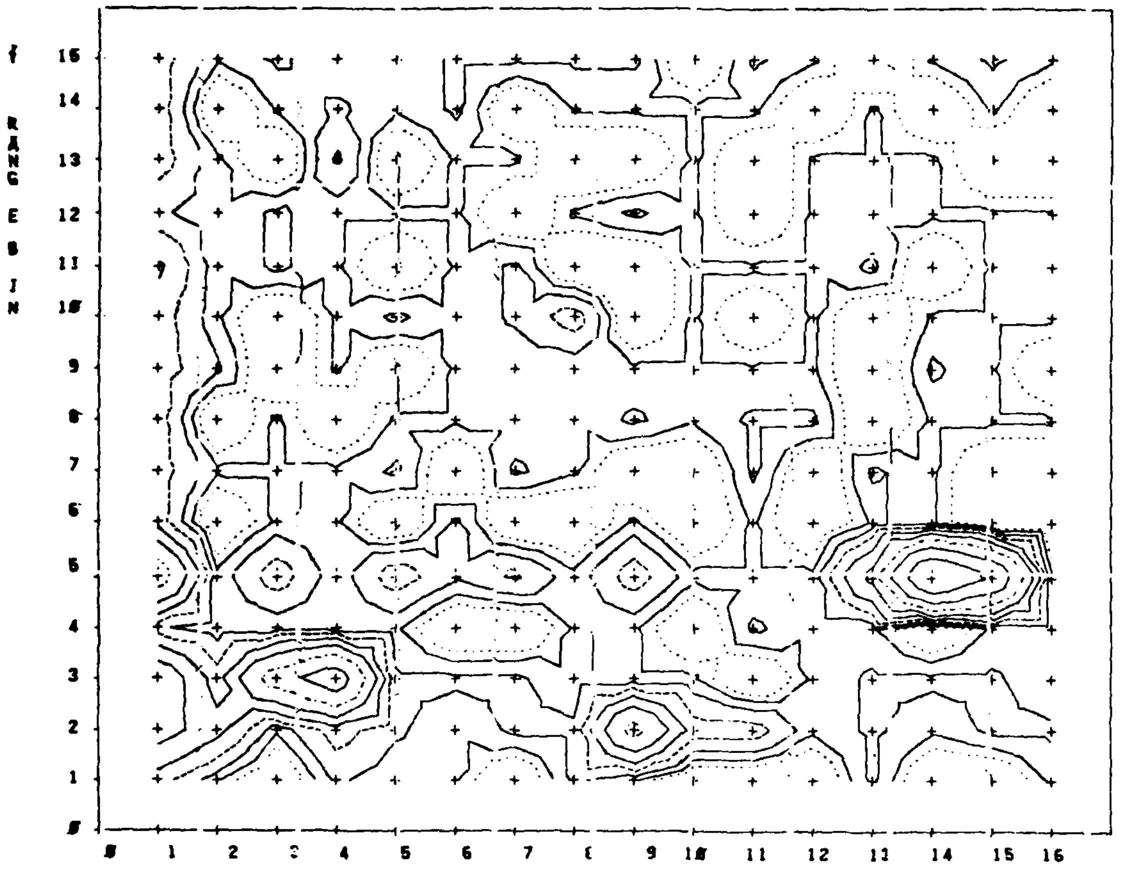


```
LOGARITHM  PROCESS  DATA
( WINDOW         CALCULATION
 BITS    FRAC     MANT       EXP
  6        3        6          0

   SIDELOBES(DB)      TYPE
     40.0               2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A9.

```
PRCDUCED BY '+WI.S:K-VT2A' ON 27OC183 AT 15.15.59      1


                    RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTFUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS

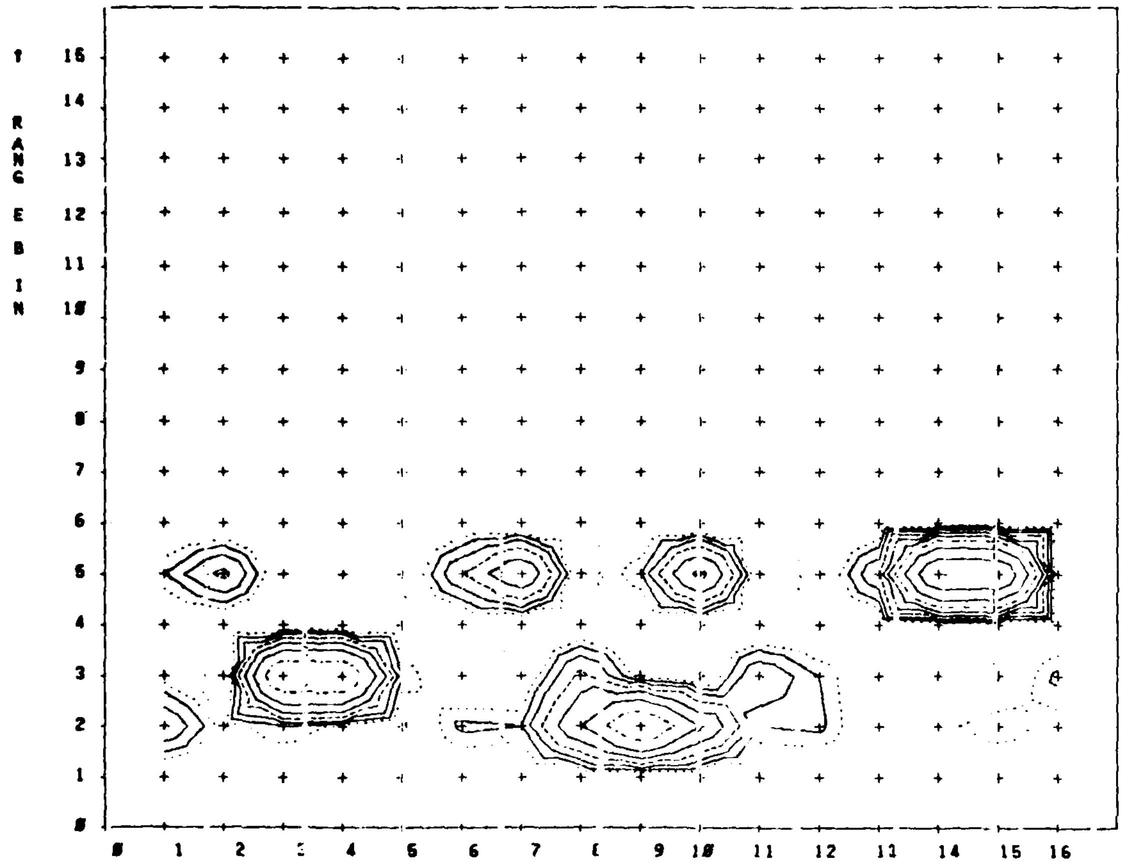


```
WINDOW DATA
  WINDOW          CALCULATION
 MANT    EXP      MANT      EXP
  5       0         5        0

   SIDELOBES(DB)        TYFE
      40.0               2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR


FFTDATA
           CALCULATION
  MANT(INT)        EXP(INT)
      5                -1

           FFT WEIGHTS
   MANT(INT)       EXP(INT)
       5               -1
```

Figure A10

```
PRODUCED BY ':W1.S:K-VT2A' ON 26OCT83 AT 15.6 .4      1


                    RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTPUT
```

CONTOUR HEIGHTS ARE 3DB INTERVALS



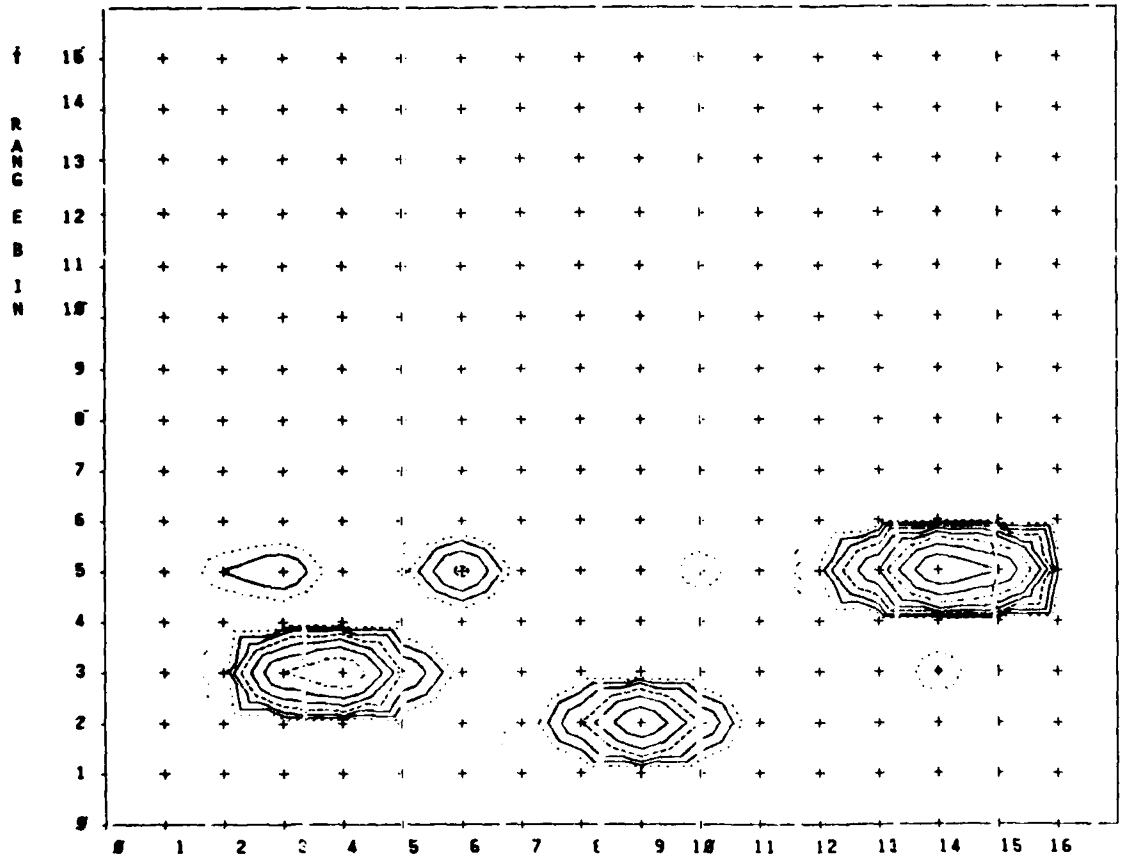
```
LOGARITHM  PROCESS  DATA
   WINDOW        CALCULATION
 BITS    FRAC     MANT      EXP
  5       Ø        6         Ø

   SIDELOBES(DB)      TYPE
       4Ø.Ø             2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A11

```
PRCDUCED B\ ':W1.S:K-VT2A' ON 26OC183 AT I!.(  34        1
```

```
RADAR HARDWARE SIMULATION   +  SIGNAL PROCESSING OUTFUT
```

```
CONTOUR HEIGHTS ARE 3DB INTERVALS
```



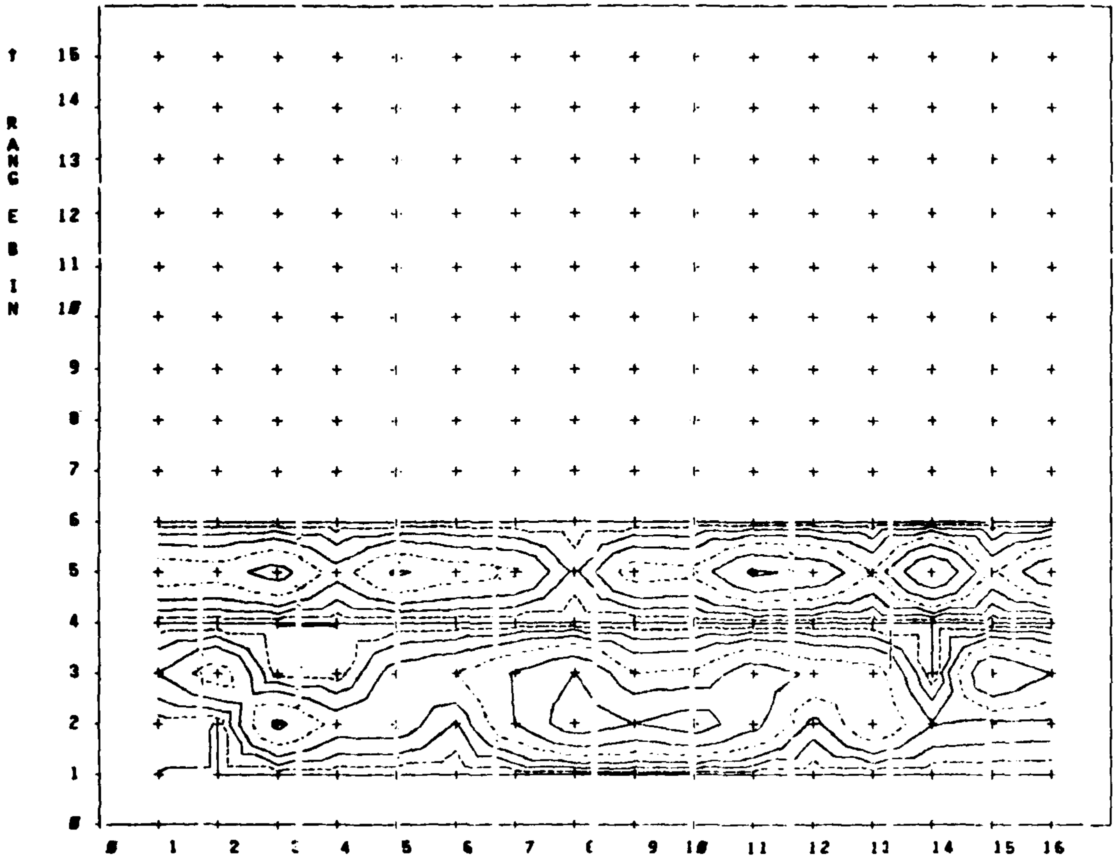
```
LOGARITHK  PROCESS  DATA
   WINDOW          CALCULATION
 BITS   FRAC     MANT       EXP
  5       1        6          0

   SIDELOBES(DB:      TYPE
       40.0              2

1-UNIFORK
2-HAMMINC
3-DOLPHCFEBY
4-KAISER
5-TAYLOR
```

Figure A12

```
PRODUCED BY ':W1.S:K-VT2A' ON 26OC183 AT 1!.:  :6        1
```

RADAR HARDWARE SIMULATION - SIGNAL PROCESSING OUTPUT

CONTOUR HEIGHTS ARE 3DB INTERVALS



```
LOGARITHM  PROCESS  DATA
  WINDOW        CALCULATION
 BITS    FRAC     MANT      EXP
  5       2        5         0

   SIDELOBES(DB)      TYPE
      40.0              2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A13

PRCDUCED BY '·WI.S:K-VT2A' ON 27OC183 AT 1Σ.1 .1 1

RADAR HARDWARE SIMULATION -· SIGNAL PROCESSING OUTPUT

ΓO. OF BINS : 15
ΓO. OF PULSES : 16
P.R.I. 1ØØ.ØØØ US

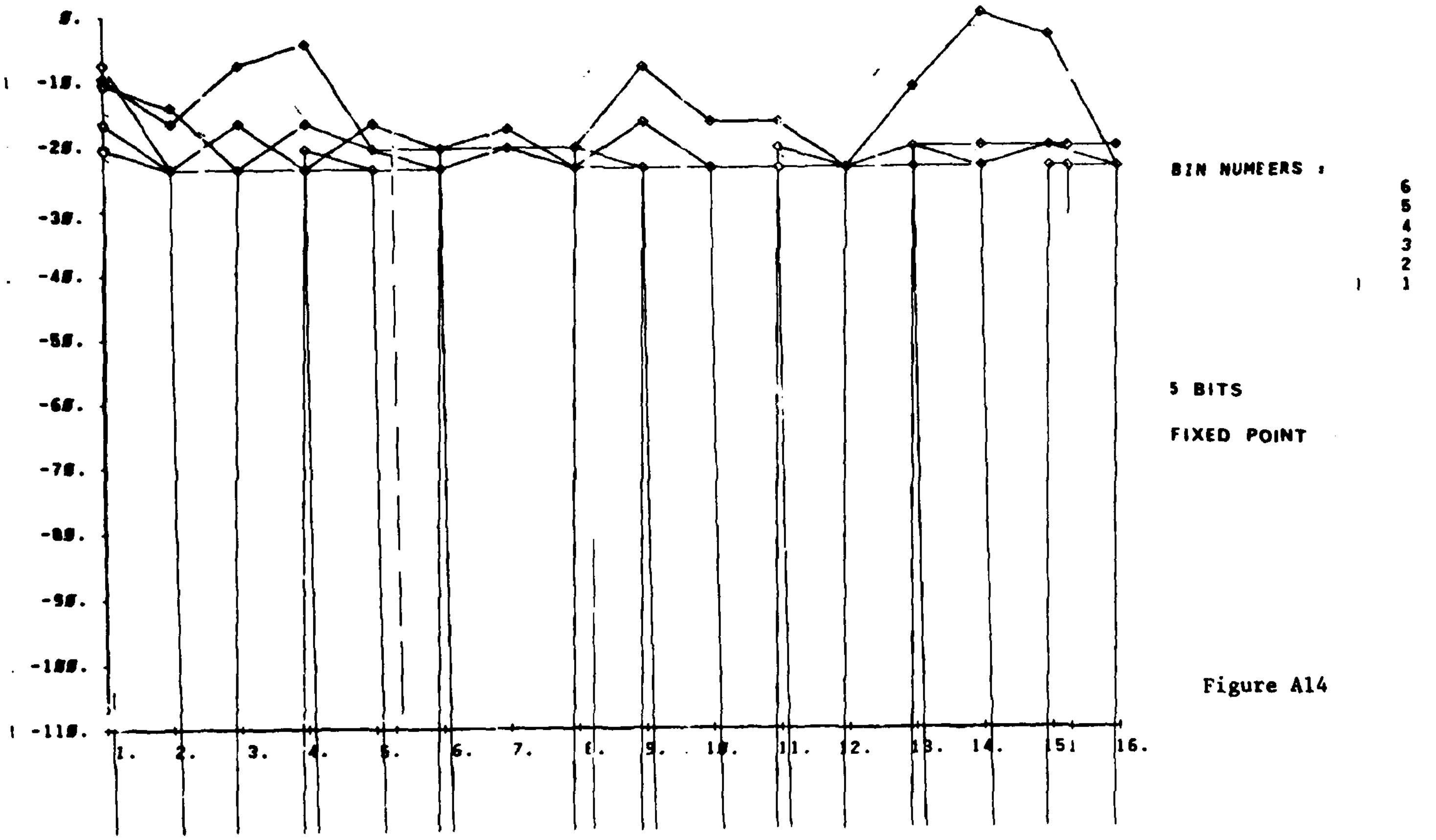
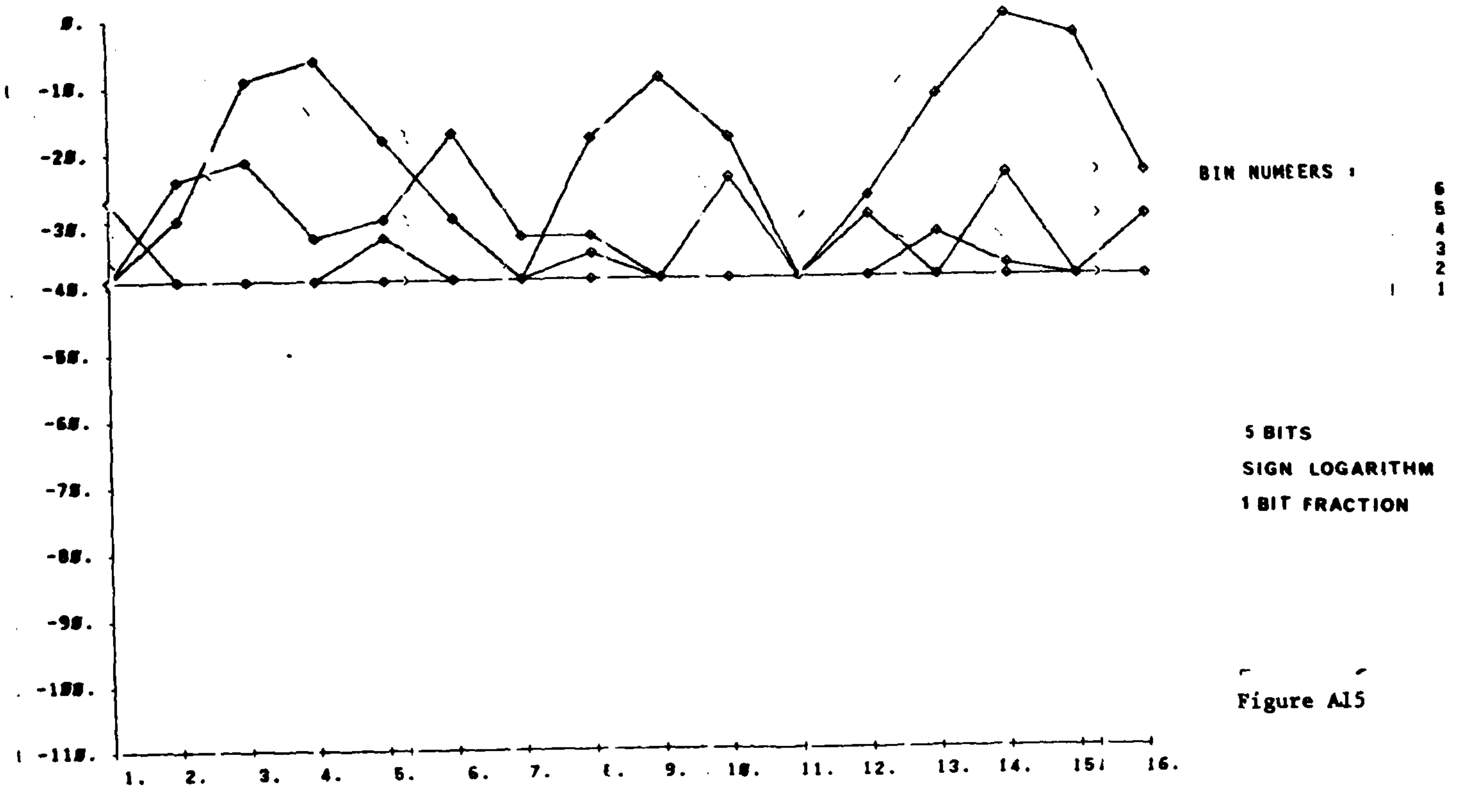


Figure A14

PRODUCED BY 'IWI.S.K-VT2A' ON 27OCT83 AT 14.4 .6 1

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

NO. OF BINS : 15
NO. OF PULSES : 16
P.R.I. : 1ØØ.ØØØ US



5 BITS
SIGN LOGARITHM
1 BIT FRACTION

Figure A15

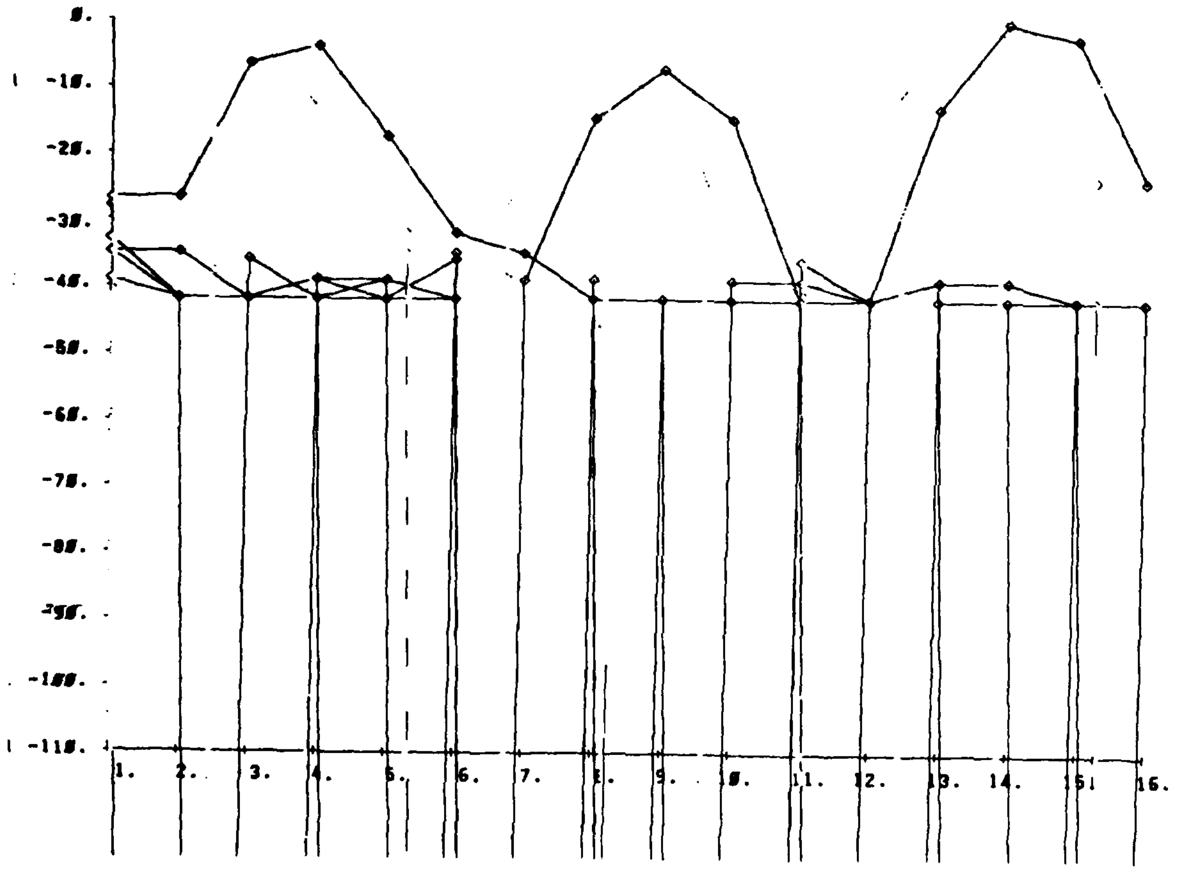
PRODUCED BY 'TWI.SCK-VT2A' ON 27OCT83 AT 15.0 .6 2

RADAR HARDWARE SIMULATION -> SIGNAL PROCESSING OUTPUT

NO. OF BINS : 15

NO. OF PULSES : 16

P.R.I. : 1ØØ.ØØØ US



BIN NUMBERS : 5 4 3 2 1 6

8 BITS

FIXED POINT

Figure A16

PRODUCED BY ':WI.SCK-VT2A' ON 27OC183 AT 14.3 .1 1

RADAR HARDWARE SIMULATION -+ SIGNAL PROCESSING OUTPUT

NO. OF BINS : 15
NO. OF PULSES : 16
P.R.I. 100.000 US

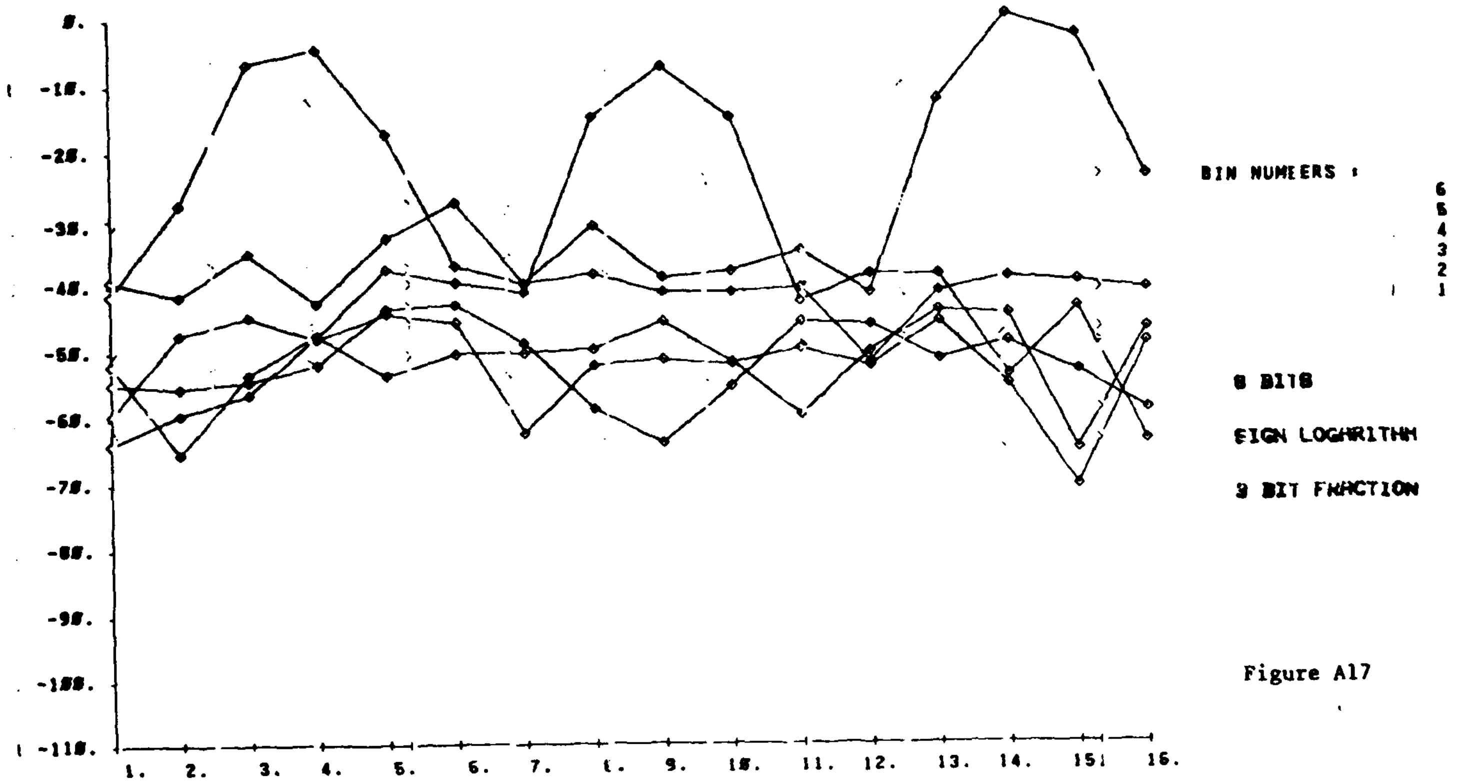


Figure A17

PRODUCED BY '.WI.S:K-VT32' ON 28OCT83 AT 1:.2..58 1

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

NO. OF BINS : 15
NO. OF PULSES : 16
P.R.I. 100.000 US

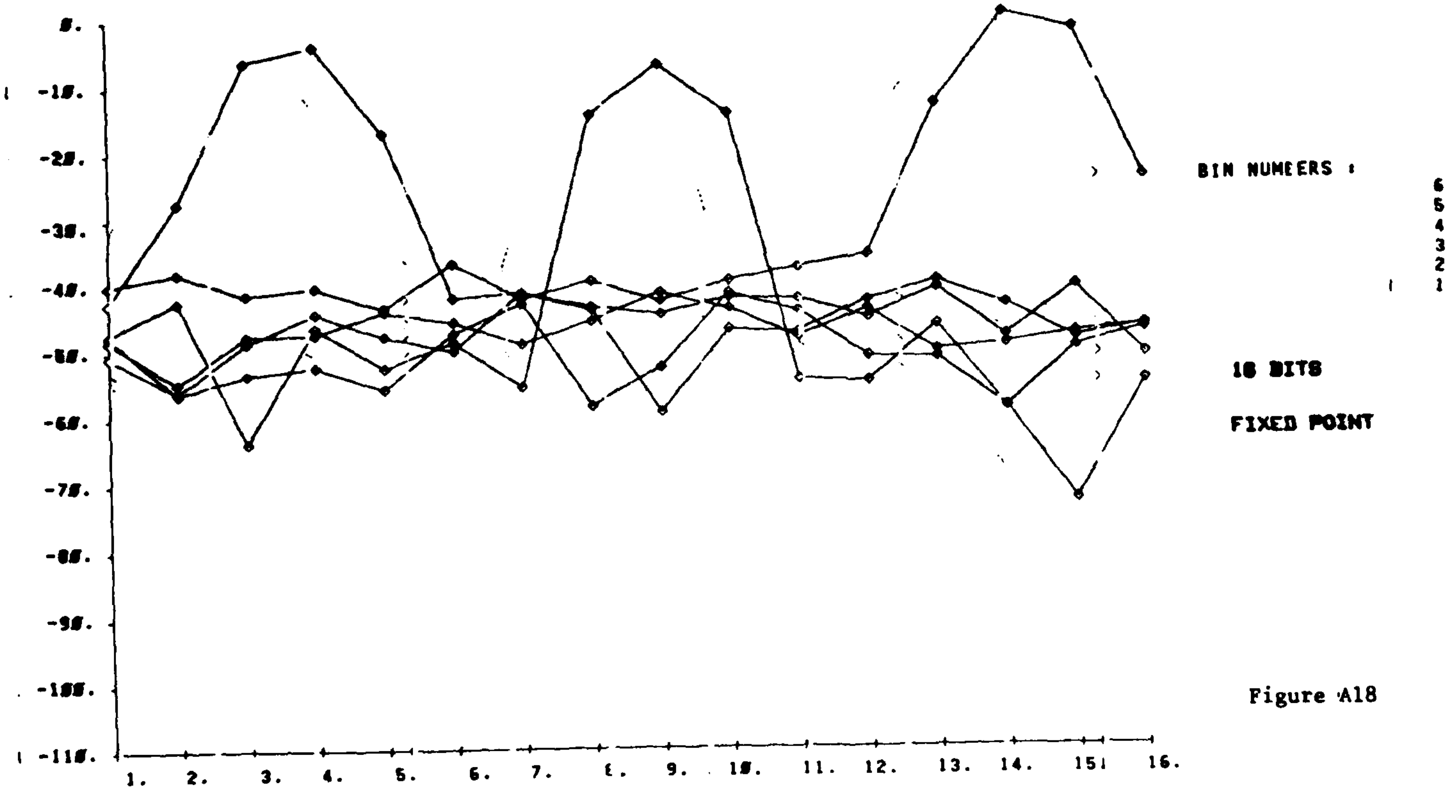
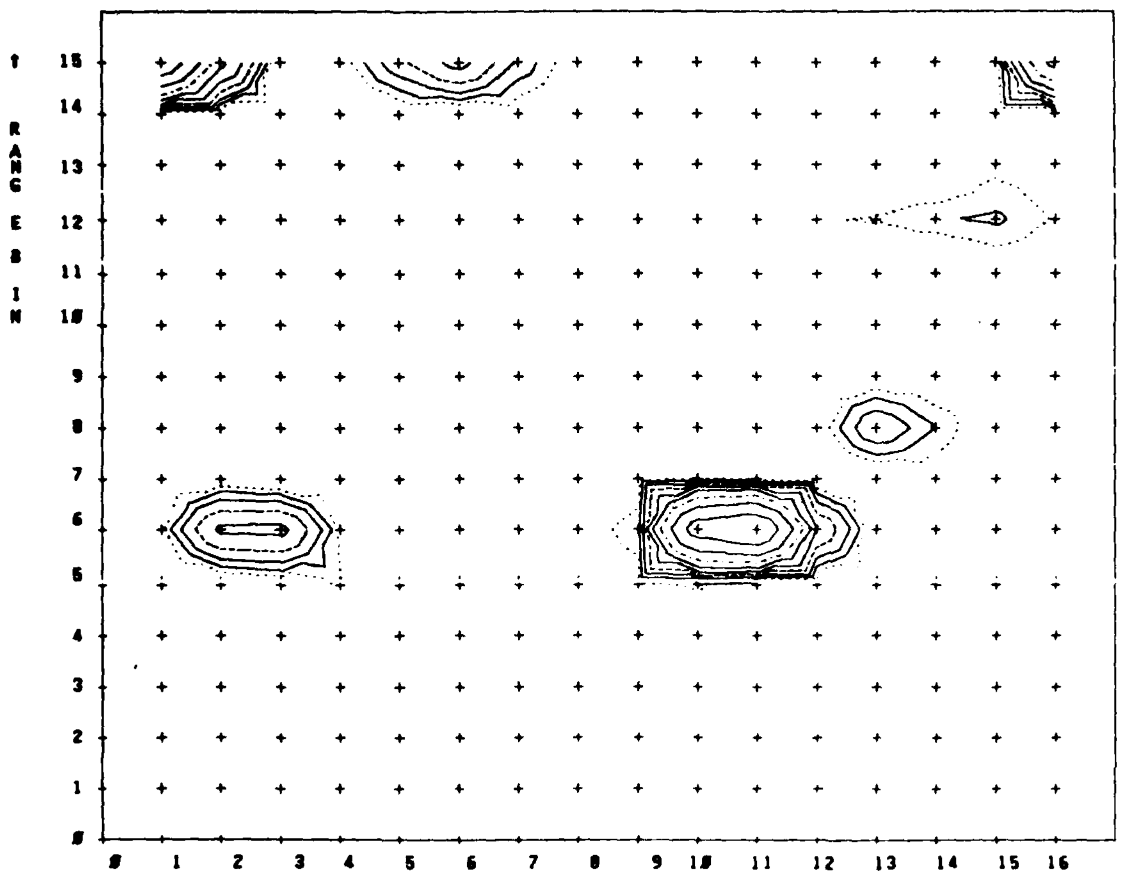


Figure A18

```
PRODUCED BY ':W1.SJK-VTZE' ON  1NOV83 AT 16.4  .4       1
```

```
RADAR HARDWARE SIMULATION   -  SIGNAL PROCESSING OUTPUT
```

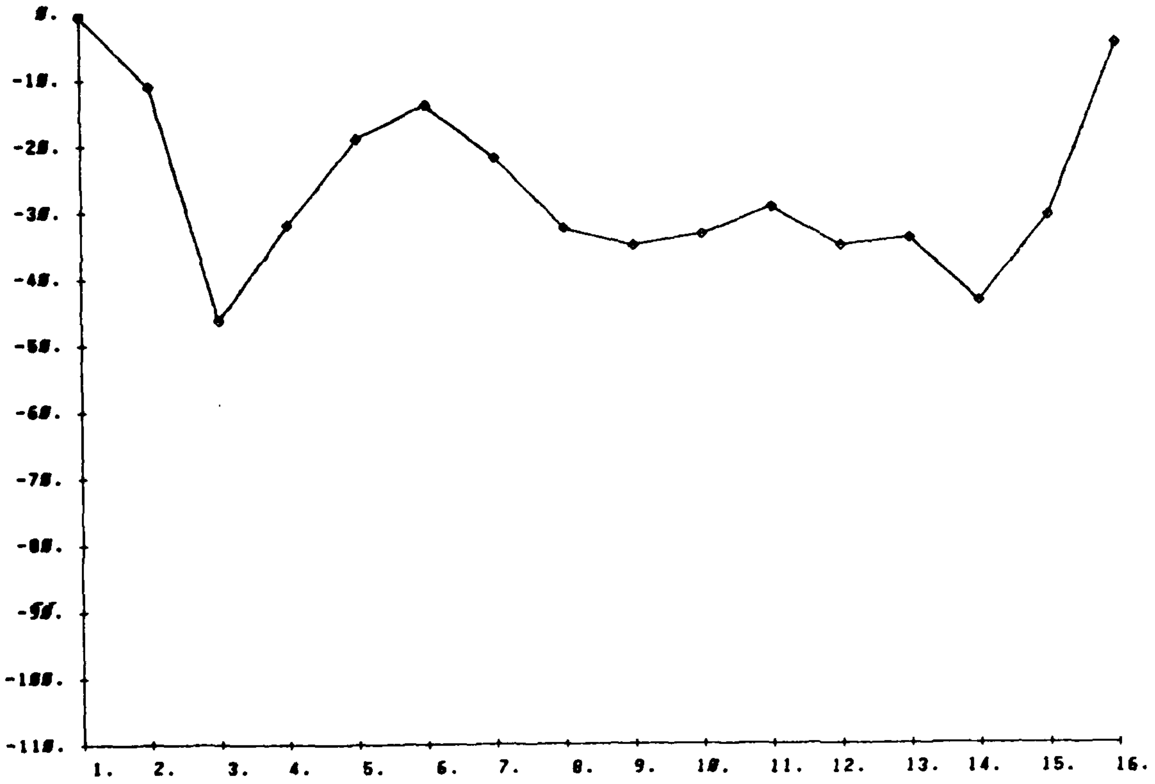
```
CONTOUR HEIGHTS ARE 3DB INTERVALS
```



```
WINDOW DATA
  WINDOW          CALCULATION
 MANT    EXP      MANT      EXP
 16       Ø        16        Ø

   SIDELOBES(DB)       TYPE
      4Ø.Ø              2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR


FFTDATA
          CALCULATION
  MANT(INT)        EXP(INT)
    16               -1

          FFT WEIGHTS
   MANT(INT)       EXP(INT)
     16              -1
```

Figure A19

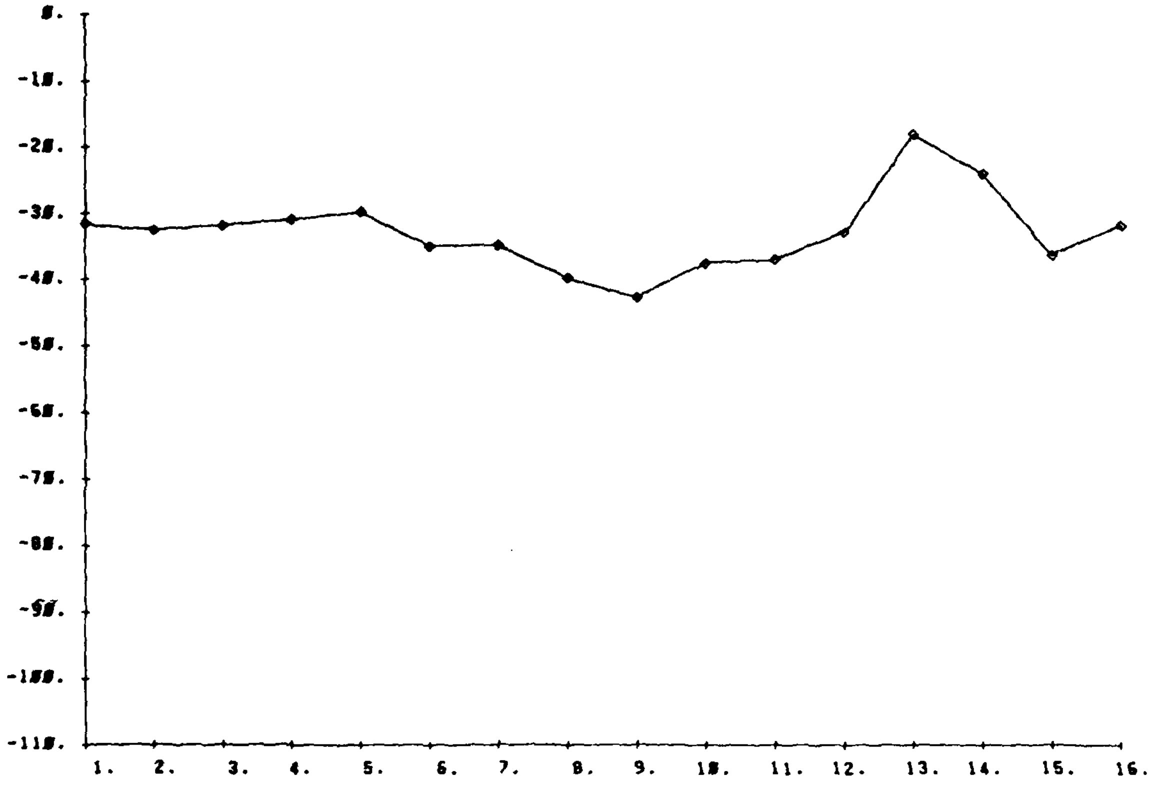
```
PRODUCED BY '›W1.SJK-VT2E' ON  1NOV83 AT 16.5  46      1


                      RADAR HARDWARE SIMULATION  -- SIGNAL PROCESSING OUTPUT
     NO. OF BINS         15
     NO. OF PULSES       16
     P.R.I.             1ØØ.ØØØ US
```



```
BIN NUMBER          16
 16 BITS
 FIXED POINT
```

Figure A20

```
PRODUCED BY '.W1.SJK-VT2E' ON  1NOV83 AT 16.5  .6      2

                    RADAR HARDWARE SIMULATION  -- SIGNAL PROCESSING OUTPUT
     NO. OF BINS         15
     NO. OF PULSES       16
     P.R.I.             1ØØ.ØØØ US
```



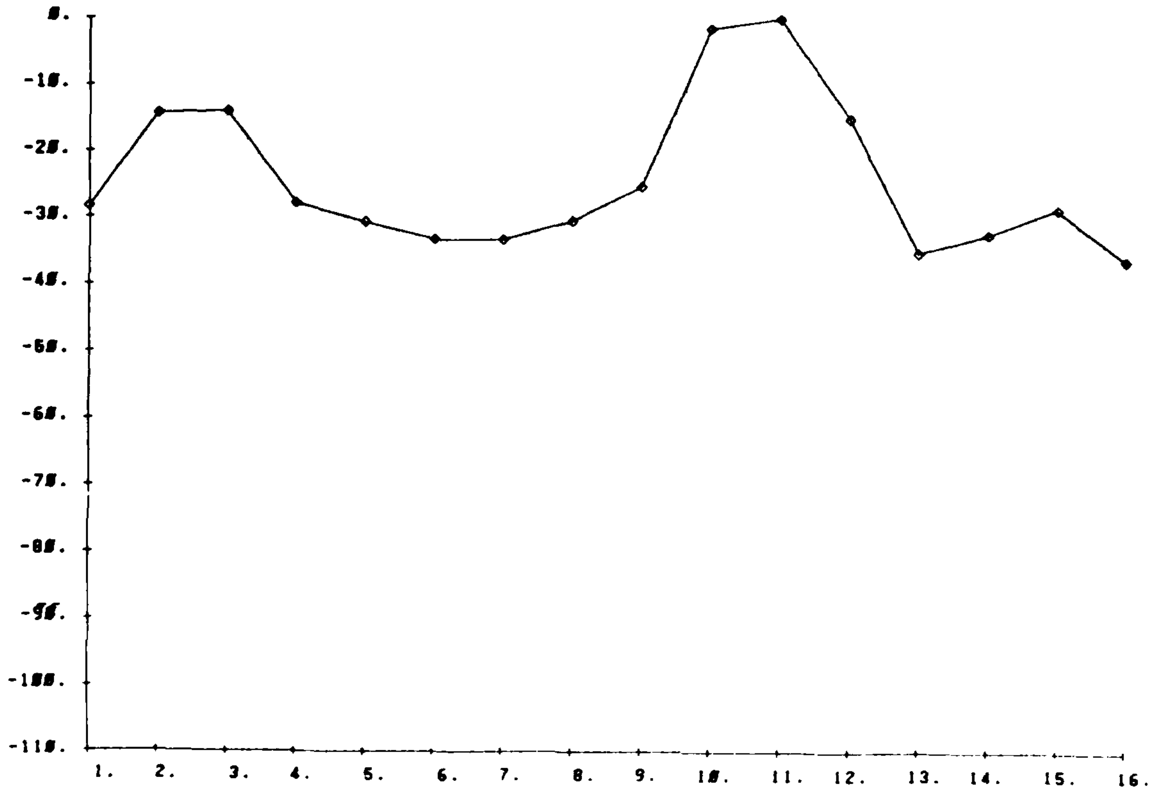
```
BIN NUMBER          8
 16 BITS
 FIXED POINT
```

Figure A21

PRODUCED BY ':W1.SJK-VT2E' ON 1NOV83 AT 16.5. 46 3

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

```
NO. OF BINS        15
NO. OF PULSES      16
P.R.I.             1ØØ.ØØØ US
```



BIN NUMBER 6

16 BITS

FIXED POINT

Figure A22

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

```
NO. OF BINS         15
NO. OF PULSES       16
P.R.I.            1ØØ.ØØØ US
```

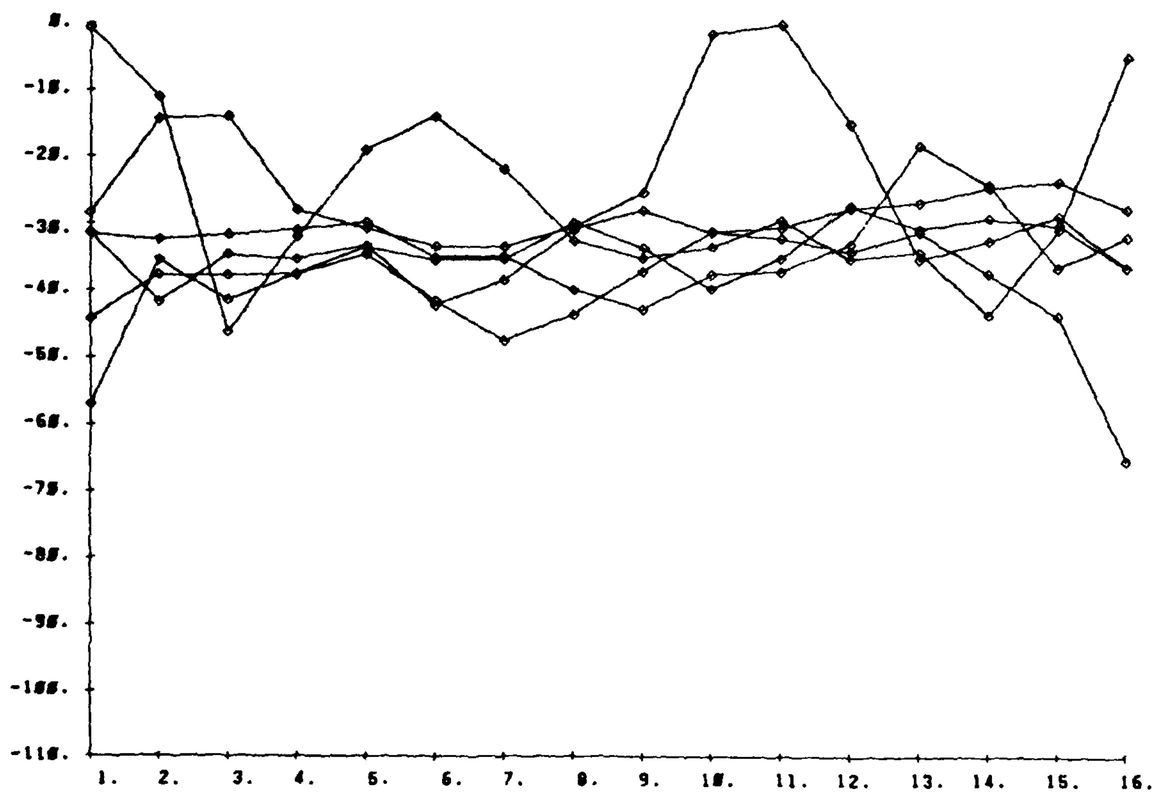
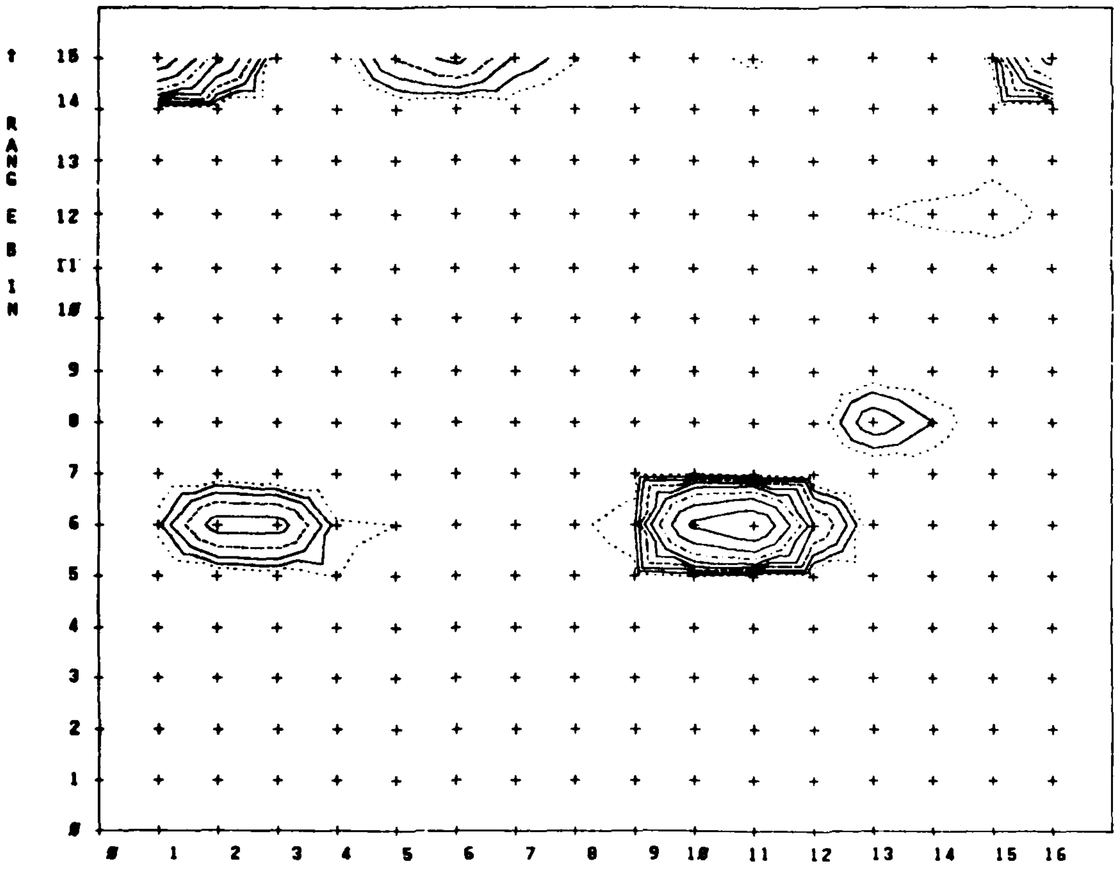


Figure A23

```
PRODUCED BY ':W1.SJK-VT2E' ON  1NOV83 AT 16.2  52        1
```

RADAR HARDWARE SIMULATION   - SIGNAL PROCESSING OUTPUT

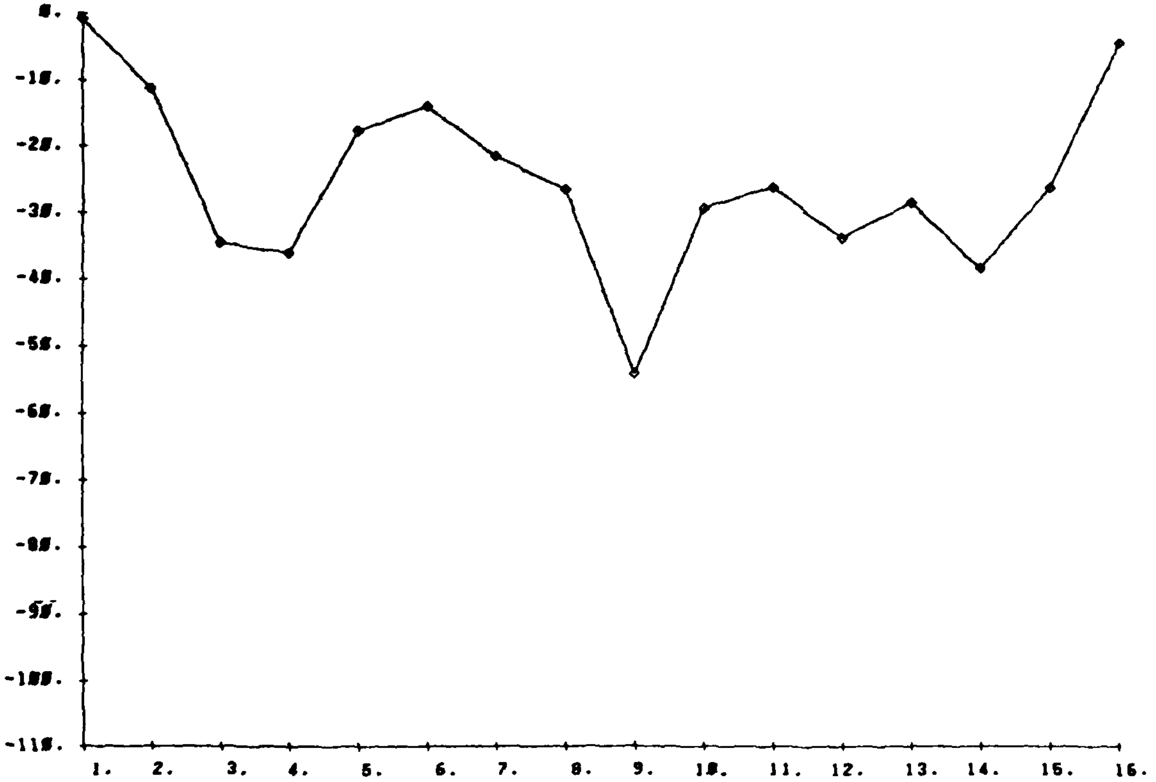
CONTOUR HEIGHTS ARE 3DB INTERVALS



```
LOGARITHM  PROCESS  DATA
  WINDOW        CALCULATION
 BITS   FRAC    MANT     EXP
  8      3       8        Ø

   SIDELOBES(DB)     TYPE
       4Ø.Ø            2

1-UNIFORM
2-HAMMING
3-DOLPHCHEBY
4-KAISER
5-TAYLOR
```

Figure A24

```
PRODUCED BY ':W1.SJK-VT2E' ON  1NOV83 AT 16.3..55       3

                    RADAR HARDWARE SIMULATION  -- SIGNAL PROCESSING OUTPUT
NO. OF BINS          15
NO. OF PULSES        16
P.R.I.              1ØØ.ØØØ US
```



```
BIN NUMBER            15
  8 BITS
  5 BIT LOGARITHM
  3 BIT FRACTION
```

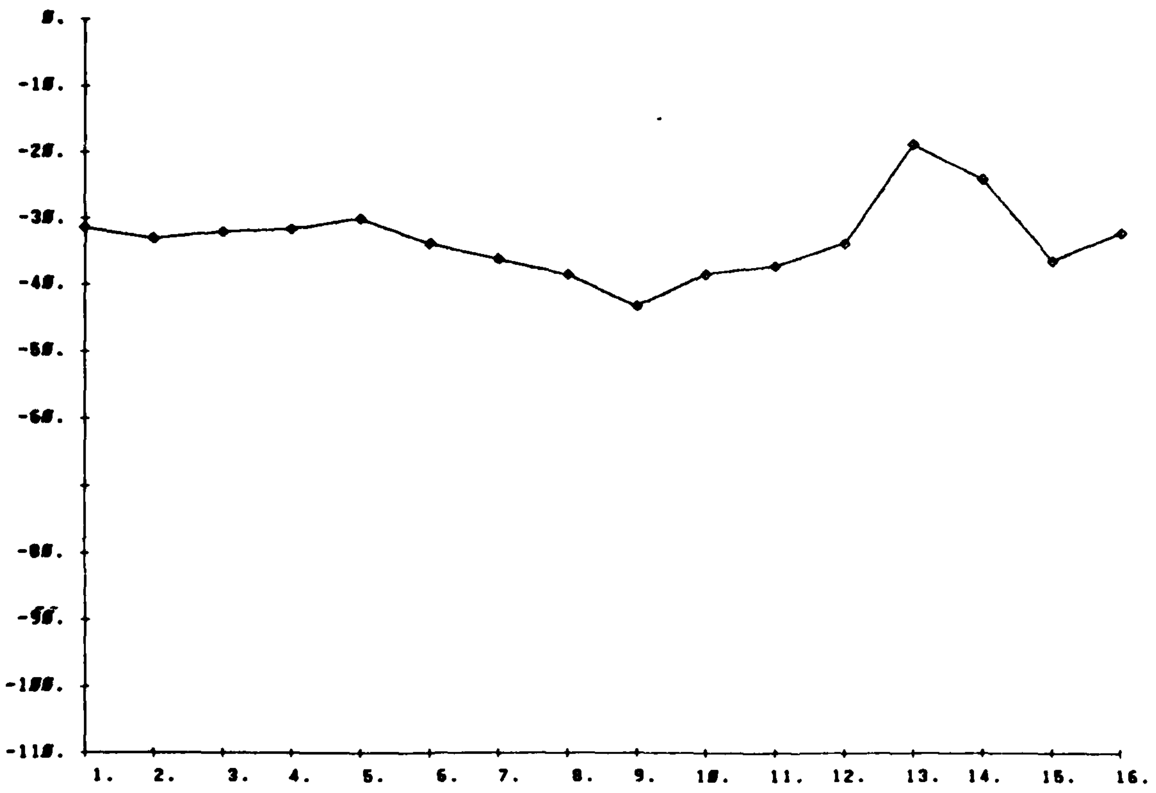
Figure A25

```
PRODUCED BY ':W1.SJK-VT2E' ON  1NOV83 AT 16.3  35       2
```

8 bit s/l 3 bit fraction

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

```
NO. OF BINS         15
NO. OF PULSES       16
P.R.I.             100.000 US
```



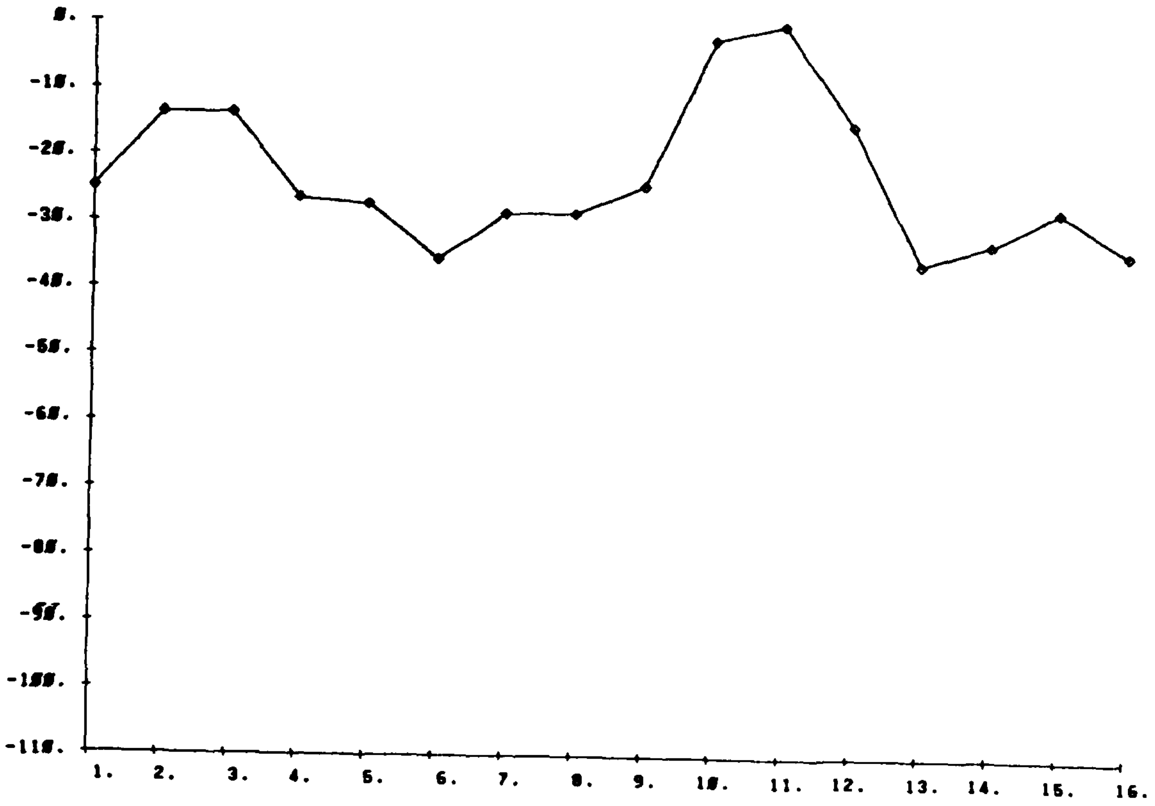
```
BIN NUMBER            8
  8 BITS
  SIGN LOGARITHM
  3 BIT FRACTION
```

Figure A26

PRODUCED BY '.WI.SJK-VT2E' ON 1NOV83 AT 16.3. 35 1

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

```
NO. OF BINS         15
NO. OF PULSES       16
P.R.I.             1ØØ.ØØØ US
```



BIN NUMBER 6

8 BITS

SIGN LOGARITHM

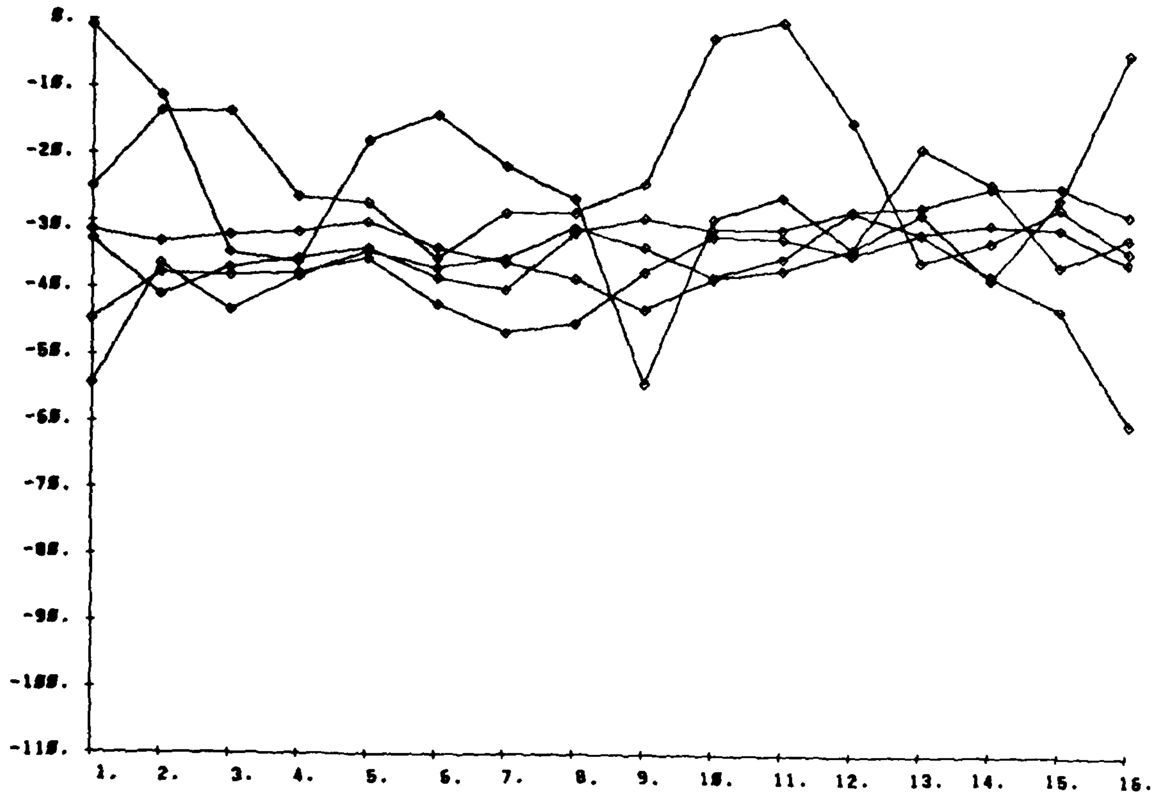
3 BIT FRACTION

Figure A27

[illegible] 3/6 5 bit fraction

RADAR HARDWARE SIMULATION -- SIGNAL PROCESSING OUTPUT

NO. OF BINS 15
NO. OF PULSES 16
P.R.I. 1ØØ.ØØØ US



BIN NUMBERS :
15
8
6
12
2
1

8 BITS

SIGN LOGARITHM

3 BIT FRACTION

Figure A28

9 REFERENCES

1 IEEE Transactions on Computers. June 1983
Vol C-32, No 6 pp 526-534
'Sign-logarithm arithmetic for FFT implementation'

2 IEEE Transactions, Acoustics, Speech, Signal Processing. Dec 1980
Vol ASSP-28 pp 706-715
'Error analysis of recursive digital filters implemented with logarithmic number systems'

3 Electronics Letters. Vol 7 1971 pp 56-58
'Digital filtering using logarithm arithmetic'

4 DR Dobbs Journal. Vol 5, No 10 Dec 1980
'N-logs, a new number language for scientific computers'
K Simons

5 'Signal Processing Modulation and Noise'
J A Betts
ISBN 0 340 09895 3
p 139 ff

DOCUMENT CONTROL SHEET

Overall security classification of sheet UNCLASSIFIED

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference MEMORANDUM 3644	3. Agency Reference	4. Report Security Classification U/C

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location ROYAL SIGNALS AND RADAR ESTABLISHMENT
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

IMPLEMENTATION OF THE SIGN-LOGARITHM ARITHMETIC FFT

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials KIDD, S J	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract

The simulation studies described show that sign-logarithm arithmetic can be implemented in a practical digital FFT analyser. Sign-logarithm arithmetic allows a smaller wordlength than conventional fixed point arithmetic whilst maintaining performance.

Discussion of the hardware implementation of such a sign-logarithm FFT shows that power consumption can be less than conventional methods using bipolar multipliers. The use of a smaller wordlength allows a significant simplification of the system into which the FFT analyser is placed and a higher data throughput rate.

END

DATE

FILMED

4 - 84

DTIC